

Volume 4

Pages 648 - 903

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC.,	)
	)
Plaintiff,	)
	)
VS.	) No. C 10-3561 WHA
	)
GOOGLE, INC.,	)
	)
Defendant.	) San Francisco, California
	) April 19, 2012

TRANSCRIPT OF JURY TRIAL PROCEEDINGS

APPEARANCES:

**For Plaintiff:**

MORRISON & FOERSTER  
755 Page Mill Road  
Palo Alto, California 94304

BY: **MICHAEL A. JACOBS, ESQUIRE**  
**KENNETH A. KUWAYTI, ESQUIRE**  
**MARC DAVID PETERS, ESQUIRE**  
**DANIEL P. MUINO, ESQUIRE**

BOIES, SCHILLER & FLEXNER  
333 Main Street  
Armonk, New York 10504

BY: **DAVID BOIES, ESQUIRE**  
**ALANNA RUTHERFORD, ESQUIRE**

(Appearances continued on next page)

*Reported By: Katherine Powell Sullivan, RPR, CRR, CSR #5812  
Debra L. Pas, RMR, CRR, CSR #11916  
Official Reporters - U.S. District Court*

*Katherine Powell Sullivan, CSR, CRR, RPR  
Debra L. Pas, CSR, CRR, RMR  
Official Reporters - US District Court - 415-794-6659*

**APPEARANCES (CONTINUED):**

**For Plaintiff:** BOIES, SCHILLER & FLEXNER  
1999 Harrison Street, Suite 900  
Oakland, California 94612  
BY: WILLIAM FRED NORTON, ESQUIRE  
STEVEN C. HOLTZMAN, ESQUIRE

ORACLE AMERICA, INC.  
500 Oracle Parkway  
Redwood Shores, California 94065  
BY: ANDREW C. TEMKIN, CORPORATE COUNSEL  
DORIAN DALEY, GENERAL COUNSEL

**For Defendant:** KEKER & VAN NEST  
633 Battery Street  
San Francisco, California 94111-1809  
BY: ROBERT ADDY VAN NEST, ESQUIRE  
CHRISTA MARTINE ANDERSON, ESQUIRE  
DANIEL PURCELL, ESQUIRE  
MICHAEL S. KWUN, ESQUIRE

KING & SPALDING LLP  
1185 Avenue of the Americas  
New York, New York 10036-4003  
BY: BRUCE W. BABER, ESQUIRE

GOOGLE, INC.  
1600 Amphitheatre Parkway  
Mountain View, California 94043  
BY: RENNY HWANG, LITIGATION COUNSEL

**For Dr. Kearl:** FARELLA BRAUN & MARTEL LLP  
235 Montgomery Street, 30th floor  
San Francisco, California 94104  
BY: JOHN L. COOPER, ESQUIRE

**Also Present:** SAFRA CATZ, President and CFO  
Oracle Corporate Representative  
  
CATHERINE LACAVERA  
Google Corporate Representative

— — —

## PROCEEDINGS

APRIL 19, 2012

7:27 a.m.

(Proceedings held in open court, outside  
the presence and hearing of the jury.)

**THE COURT:** All right. We have a few things I have to take up with you.

We have a note from one juror. Question:

"Are the APIs in question standard Oracle

APIs? Are there copyrights on these APIs in question?"

12 The way we deal with this is when the witnesses come,  
13 you can ask the appropriate witnesses these questions, but  
14 we're not going to give a new speech to the jury yet on these  
15 points.

16 Now, I may tomorrow, if time permits, give each side  
17 ten minutes to make a closing or opening statement to the jury  
18 to let them know where you stand. If you don't want to use it,  
19 you don't have to. But sometimes it helps in a case like this  
20 to let the lawyers wax eloquent in front of the jury to explain  
21 where we are, what's been proven, what's not been proven. I  
22 think it helps the jury comprehend the evidence as we go along.

23                   In the meantime, if you have a witness who can answer  
24 this question, these two questions, you're free to go for it.

Okay, next item. My law clerk -- where is my law

1 clerk -- will hand out to some questions for a special verdict  
2 form.

3 (Whereupon document was tendered  
4 to counsel.)

5 This is definitely a draft, and here is what I would  
6 like for you to do by tomorrow morning is to do in handwriting  
7 -- do this in handwriting so I can clearly see what it is you  
8 want to change and then do a short statement that explains why  
9 you want the change. I'm trying to come up with the most  
10 streamlined version I can of what the special verdict form  
11 would look like.

12 Now, a related question is: How do affirmative  
13 defenses fit in? It's possible that we would just let the  
14 affirmative defenses go to the jury anyway, the implicit parts  
15 of these liability questions.

16 On the other hand, maybe we want to have separate  
17 questions on the affirmative defenses. But I haven't addressed  
18 that per se, and I invite you to do so by tomorrow morning. So  
19 that's item number two.

20 Item number three is that at about 9:00 o'clock today  
21 I need to take the 15-minute break a little early in order for  
22 me to be on a telephone call. So I may have abruptly stop  
23 proceedings when that occurs, but it's best to do it. So we'll  
24 take our normal break at that time.

25 Next. On the deposition designations for Mr. Gupta

1 and Mr. Risvi, would somebody from Google come up here and take  
2 these?

3 (Whereupon documents were tendered  
4 to counsel.)

5 **THE COURT:** I have indicated in the margin what the  
6 rulings are, and I hope you can decipher what I've indicated.  
7 Be sure to share that with the other side.

8 One rule of evidence that you all, I'm sure, know is  
9 that you can never, never read deposition testimony about a  
10 document that is not in evidence. And I have seen cases where  
11 the document doesn't come into evidence and, therefore, the  
12 testimony doesn't come in even though the lawyer, if they had  
13 done it right, could have avoided the problem by asking the  
14 question independently of the document.

15 This is a good example of where many, many, many  
16 questions are tied into an email. Now, I'm assuming the email  
17 is going to come into evidence, but you didn't tell me that in  
18 your write-up. So I had to -- you need to let me know if there  
19 are objections to the email or whatever it is that -- because  
20 if the email doesn't come in, the testimony doesn't come in.

21 Because so many of these we're reading from -- for  
22 example, it would say, "In paragraph number two the email says  
23 X, Y, Z. Tell me what you know about that." Well, you can't  
24 have -- if that's not in evidence, you're not going to be  
25 allowed to read that testimony. That's a fundamental rule of

1 evidence.

2 So I'm assuming that's not a problem because you  
3 surely have figured that out and know that that email has got  
4 to come into evidence some other way, but it was not clear from  
5 the write-up that that was true. So as we go forward doing  
6 these things, help me out on that piece of it by explaining  
7 whether or not there is an issue over the admissibility of the  
8 document that the deposition is about.

9 All right. My next item. Do we have the one-page  
10 scorecard with all deposition exhibits and trial exhibits, or  
11 is it system in process?

12 **MR. JACOBS:** Still in process, your Honor.

13 **THE COURT:** A list of Who's Who? Is that in process?

14 **MR. JACOBS:** We prepared a list and sent it over to  
15 Google and asked them to look at it.

16 **THE COURT:** All right. Thank you.

17 Now, I have one last item. Two more -- three  
18 questions more from the jury. I will read them out loud and  
19 you can see them in writing soon.

20 "Seems Sun was aware that Apache had  
21 fragmented" -- oh, I misread that.

22 "Seems Sun was aware Apache had fragmented  
23 Java and was offering potentially  
24 non-compatible APIs, etc that others may have  
25 downloaded for future use. What formal

1 communication did Sun have (i.e., revoke  
2 license? Have Apache stop distributing?  
3 Contact those who acquired Apache to caution  
4 about use?) If not, why not? If so, when?"  
5 So that's that question.

6 Next question.

7 "It was testified that Apache was shelved by  
8 who" -- no.

9 "It was testified that Apache was shelved -  
10 when? Why?"

11 "What are/were terms of Apache license?"

12 "Did Google get Apache license? If so, when?  
13 (2005?)"

14 "Were 37 APIs available - Apache license?"

15 Then the final question.

16 "How (by what means) do the lawyers for one  
17 company obtain the private information or  
18 documents, i.e., meeting reports, emails,  
19 et cetera, with which they confront witnesses  
20 of the other company?"

21 Well, that's one I will answer. So our jury is  
22 generating questions. Good, good, good.

23 All right. I will let you see all of those  
24 originals. Did we hand those out already?

25 **THE CLERK:** They all have copies.

1                   **THE COURT:** All right. You already have copies.

2                   Now, what are your issues?

3                   **MR. JACOBS:** Your Honor, we have reached agreement to  
4 some exhibits to move into evidence. If I can do that, I would  
5 like to.

6                   **THE COURT:** You may, but let me get my chart. Go  
7 ahead.

8                   **MR. JACOBS:** 36, 37, 38, 39, and 40.

9                   **THE COURT:** Agreed?

10                  **MR. VAN NEST:** No objection, your Honor.

11                  **THE COURT:** All received.

12                  (Trial Exhibits 36, 37  
13                   , 38, 39 and 40 received  
14                   in evidence)

15                  **MR. JACOBS:** 41, 42, 43, 44, 45, 46.

16                  **THE COURT:** Agreed?

17                  **MR. VAN NEST:** No objection, your Honor.

18                  **THE COURT:** Received.

19                  (Trial Exhibits 41, 42, 43, 44, 45, 46 received  
20                   in evidence)

21                  **MR. JACOBS:** 47, 51, 53, 54, 65, 70, 74.

22                  **MR. VAN NEST:** No objection.

23                  **THE COURT:** All received.

24                  (Trial Exhibits 47, 53, 54, 65, 70, 74 received  
25                   in evidence)

1           **MR. JACOBS:** 30.

2           **MR. VAN NEST:** No objection.

3           **THE COURT:** Received.

4           (Trial Exhibit 30 received  
5           in evidence)

6           **MR. JACOBS:** There are additional exhibits as to  
7 which we have not been able to secure Google's agreement to  
8 admissibility, notwithstanding that they were in the particular  
9 case that I'm looking at, cases that I'm looking at, downloaded  
10 from Google's website.

11           It is possible that given the kerfuffle yesterday  
12 about authentication and admissibility, that we will have no  
13 choice but to put an associate on the stand to secure the  
14 admission of those exhibits.

15           What I would ask the Court to do is under those  
16 circumstances reserve the question of whose time that should be  
17 counted against so that if we are put to the use of our  
18 valuable trial time by having to put an associate on the stand  
19 to confirm that that which is on Google's website is that which  
20 is in front of us now, when we have sent over the exhibit to  
21 Google to ask, to confirm that very point, that we would ask  
22 that that time be charged to Google's time and not Oracle's  
23 time.

24           I hope that we'll be able to work this out.

25 Mr. Van Nest and I have talked about the importance on our side

1 of giving them a heads-up of what we want to admit and giving  
2 them a chance to look at it. He has assured me that if it is  
3 what -- if the document is what it purports to be and it comes  
4 from Google's production at least, we haven't specifically  
5 talked about this issue yet, that we will not encounter  
6 difficulties, but I needed to alert the Court of our plans.

7                 Obviously, the associate was not on our rolling  
8 witness list. We would ask for forgiveness on that account if  
9 we are put to this burden.

10                 **THE COURT:** What does Google say?

11                 **MR. PAIGE:** Well, your Honor, I would like to explain  
12 a little bit about what Mr. Jacobs is talking about.

13                 One of the things he's talking about is a version of  
14 the Android source code they say that was downloaded on  
15 March 12, 2012. That was sent over to us late at night on  
16 April the 12th, 2012.

17                 Now, I take them at their word that that's when it  
18 was downloaded. We don't know that. No one has sworn to the  
19 fact that this was downloaded on March 12, 2012.

20                 I think it would be appropriate for something that is  
21 going to be -- if it's going to be represented to have been  
22 taken from a website at a certain date, that we have some sort  
23 of testimony authenticating that to be the case.

24                 Certainly we have no objection to anything that we  
25 have produced in the case and we know what it is being put into

1 evidence and we've agreed to that. This is something that has  
2 been downloaded by them --

3           **THE COURT:** I'm sorry. The representation to you is  
4 it was downloaded when?

5           **MR. PAIGE:** March 12, 2012, and it was provided to us  
6 on April 12th, 2012.

7           **THE COURT:** Do you deny that it was available on the  
8 website on that day?

9           **MR. PAIGE:** Your Honor, it's -- it's literally  
10 probably a gigabyte of data. I assume it's available on the  
11 website that date. I don't know exactly what the website would  
12 have looked like that date or what might have changed in the  
13 interim. But I think it's important if we're going to have  
14 representation this was done on a certain date, that someone  
15 attest to it.

16           If this were a matter of something that were two  
17 pages long, it would be easy to look at it. This is something  
18 that is literally a gigabyte of information.

19           **THE COURT:** Well, on the gigabyte part, as opposed to  
20 the timing part, are you saying it what been forged or  
21 something?

22           **MR. PAIGE:** I'm not saying that at all, your Honor.  
23 I have to doubt that it is what it is. It's a question of  
24 whether there is someone that's going to say, in fact, that's  
25 what happened.

1                   **THE COURT:** You probably know good and well it was  
2 available on March 12th.

3                   **MR. PAIGE:** Your Honor, there is no doubt that  
4 something was available on March 12th. I agree with that  
5 entirely.

6                   **THE COURT:** All right. Here is the answer. The  
7 answer is Mr. Jacobs has permission to bring in one of your  
8 associates, whoever, out of turn or whenever you wish without  
9 notice, put him on the stand and go through this drill. I will  
10 reserve on the issue of whose time it comes out of, but I'm  
11 not -- I'm not saying it's going to come out of Google's time.  
12 You would have to convince me that what Google is doing is  
13 unreasonable and I'm not sure. I'm not yet sure on that point,  
14 but it's possible.

15                  On the Google side, be aware that the day will come  
16 in this trial when you will be begging the Court to excuse some  
17 goof up on your part or some other -- something that you want  
18 to do that is not in accordance with -- strictly in accordance  
19 with the guidelines and Mr. Jacobs is going to remind me about  
20 this conversation.

21                  So you should be -- this is the kind of thing usually  
22 lawyers work out in a big trial like this so that the jury can  
23 focus on things that are really in controversy and not somehow  
24 get the idea that this is in controversy. So that's the  
25 answer.

1           I'm going -- the answer is: Mr. Jacobs, you've got  
2 to do it the hard way for now.

3           **MR. PAIGE:** And, your Honor, we will continue to  
4 discuss this with them. We received this just last evening and  
5 obviously we're still...

6           **THE COURT:** You always say that, to make it sound  
7 like you're not being so unreasonable; that you're going to  
8 continue -- that sounds like international negotiations.  
9 "We're continuing to talk."

10           Well, this is something you should have figured out  
11 overnight. Come on.

12           All right. That's -- yes, Mr. Van Nest.

13           **MR. VAN NEST:** The only comment I want to make, your  
14 Honor, is that the number of things that we have already  
15 forgiven compared to the number of things that are in issue  
16 before your Honor is monumental. And I do want to say that  
17 most staffs, not just our staff, are working hard.

18           Many of these things come over late at night. They  
19 should have been done weeks ago. This was downloaded  
20 apparently in March, not given to us until April. It was long  
21 after discovery closed.

22           The stuff that's been produced in the case, no  
23 problem. It's all been handled very smoothly. This is all  
24 stuff that they are throwing in day after day, sometimes late  
25 at night, that was done all after discovery closed.

1           That's why there are a few tag-end issues and I  
2 apologize for taking the Court's time.

3           **THE COURT:** That's a good point and I understand  
4 that. But I did hear April 12. This is April 19th -- or is it  
5 20th yet? 19th.

6           **MR. VAN NEST:** The request to put these in was last  
7 night. April 12 was when he got a big data dump of this stuff.  
8 The request to put these in by stipulation was last night.

9           **MR. JACOBS:** That's not complete. We moved to add  
10 these to the exhibit list. These were called out in a motion  
11 because these are the source code files that controvert  
12 Google's claim that they have removed the code from  
13 distributions of Android. That's the history here, and your  
14 Honor saw that motion. Google saw that motion. So this has  
15 been coming as an issue.

16           They put it into play by denying that the code  
17 continued to be available on the public website. So there has  
18 been no surprise, your Honor.

19           **THE COURT:** All right. Enough said on this.

20           Let's move to the next point.

21           **MR. JACOBS:** Your Honor, I would like to propose that  
22 we keep track, at least in the Court's record, of  
23 demonstratives and so I've got -- we provided this to Google --  
24 a set of demonstratives as Trial Exhibit 1046 that have been  
25 used about Mr. Reinhold. And I would hand those to the

1 courtroom deputy.

2 (Whereupon, document was tendered  
3 to the Court.)

4 **THE COURT:** Yes. We should definitely keep track of  
5 what is a demonstrative.

6 Now, as a heads-up to the -- I say this often to the  
7 juries, but it wouldn't hurt for you to sometimes say it.  
8 There will be people on the jury who think that the  
9 demonstratives will be in the jury room with them.

10 I have had trials where the very first note during  
11 deliberations is, "Can we see Poster X?" Answer, no. And they  
12 didn't take any notes while that -- and they could have.

13 Now, sometimes I remember to say to the jury, which  
14 most judges would not even mention, but because I've seen this  
15 problem time and again: Demonstratives are not going to be --  
16 these are argument pieces and they are not going to be in the  
17 jury room.

18 So you can ask me and I will remind the jury that  
19 they are not going to be in the jury room, but I can't do it  
20 absolutely every time.

21 But you are correct. We want to have a record for  
22 the Court of Appeals of what is used as a demonstrative. So  
23 fine, this 1046.

24 What else?

25 **MR. JACOBS:** Your Honor asked some questions

1 yesterday and the way the Court framed the question was if this  
2 is going to come into evidence, I would like to know the answer  
3 to the following questions. And we did some homework and  
4 Mr. Reinhold would be prepared to testify to the answers to  
5 those questions before the jury. It would be pretty brief.  
6 We've provided these to Google as well.

7 We have summarized in the nature of summaries the  
8 answers to the Court's question. I wanted to do this out of  
9 the presence of the jury so we could get your guidance on how  
10 you would like to proceed.

11 THE COURT: Could I see what it is you have there?

(Whereupon, document was tendered  
to the Court.)

14                   **THE COURT:** The first one says 1063. What would the  
15 point here?

16                   **MR. JACOBS:** 1063 "Classes and interfaces known to  
17 the Java Language Compiler," on the upper left-hand.

18                   **THE COURT:** It says, "Classes and interfaces known to  
19 the Java Language Compiler."

20                   **MR. JACOBS:** So that answers the question if you are  
21 preparing a compiler to the Java language specification proper,  
22 what classes and interfaces in the Java.location programming  
23 interface specification must be implemented in that compiler,  
24 what classes and interfaces are part of the semantics of Java  
25 language proper programming instructions.

1                   **THE COURT:** It looks like eight of them are. Is that  
2 right? Eight of them, or all of these are?

3                   **MR. JACOBS:** That's where I will get Dr. Reinhold to  
4 be more specific, your Honor, explain what this is showing.

5                   **THE COURT:** What would be the possible objection to  
6 him explaining this?

7                   **MR. PURCELL:** We don't have an objection, your Honor.

8                   **THE COURT:** All right. So you can use 1063 and go  
9 into that subject matter.

10                  **MR. JACOBS:** And 1064 answers a somewhat different  
11 and less --

12                  **THE COURT:** 1062 it says here.

13                  **MR. JACOBS:** I'm sorry.

14                  **THE COURT:** 1062 answers what?

15                  **MR. JACOBS:** The document titled "Classes and  
16 Interfaces Mentioned in the Java Language Specification."

17                  It answers the question, if you go through the formal  
18 version created by the language specification, what classes and  
19 interfaces are shown there, excluding examples.

20                  They are not part of the semantics of the language.  
21 They are not part, obviously, of the syntax of the language,  
22 but they are -- but they are mentioned. And we do this out  
23 of -- mostly out of completeness.

24                  **THE COURT:** I see. Any objection to that part?

25                  **MR. PURCELL:** No, your Honor.

1           **THE COURT:** You may use both and explain both.

2           **MR. JACOBS:** Terrific.

3           **MR. VAN NEST:** Did we get the copies? I didn't  
4 realize they were going to be exhibits, your Honor.

5           **THE COURT:** Of course you should have copies.

6           **MR. JACOBS:** They do have copies. They just don't  
7 have exhibit numbers on them.

8           **THE COURT:** Do this. You can have my copies.

9           **MR. JACOBS:** We have copies of it.

10          **THE COURT:** Do you have copies with exhibit numbers  
11 for counsel?

12          **MR. JACOBS:** Let me retrieve those.

13          **THE COURT:** You may take these back.

14          (Whereupon document was tendered  
15           to counsel.)

16          **THE COURT:** What else?

17          **MR. JACOBS:** I believe that's all we have to raise.

18          **THE COURT:** How about Google? Any issues?

19          **MR. VAN NEST:** I don't believe so, your Honor.

20          **THE COURT:** I have up here something, a stray  
21 document. Here I'm going to hand it down to counsel.

22          (Whereupon document was tendered  
23           to counsel.)

24          **THE COURT:** I don't remember now why you handed that  
25 to me. Do I need that any more for anything? It was from

1 | yesterday.

2                   **MR. BOIES:** No, your Honor. Those were examples of  
3 documents that we offered, and the Court ruled that there was  
4 not an adequate foundation because the witness couldn't -- said  
5 he couldn't identify them.

6                   **THE COURT:** But without prejudice to doing it with  
7 somebody else and bringing that witness back.

8 || MR. BOIES: Yes, your Honor.

9 THE COURT: Yes I remember now. Okay.

10                   **MR. JACOBS:** We do have one more item Mr. Norton will  
11 address.

12 THE COURT: All right, Mr. Norton.

13                   **MR. NORTON:** During the examination of Mr. Page,  
14 Exhibit 6 was offered and published to the jury, but we don't  
15 show it as admitted and we just wanted to make sure --

16 THE COURT: I show it as admitted.

17 || MR. NORTON: Thank you, your Honor.

THE COURT: Okay.

19                   **MR. VAN NEST:** We don't have anything else, your  
20 Honor.

21                   **THE COURT:** All right. Let's bring in the jury and  
22 bring back the witness.

23 || (Jury enters courtroom at 7:53 a.m.)

24 THE COURT: Welcome back, and please be seated.

Before we resume with our -- our witness can have his

1 seat.

2 Before we resume with the witness, first I want to  
3 thank you all for being so punctual again. You all getting  
4 enough sleep and everything? Good. Everyone good on that?

5 All right. So anyone coming down with the flu? No?  
6 Good. You know, keep washing your hands. If you get sick, we  
7 may have to take a day and just wait until you get well. So I  
8 want you to stay as healthy as you possibly can.

9 Now, you remember I said you could write notes and  
10 ask questions and you -- four of you have done that already.  
11 That's good. You don't have to. Some trials nobody writes a  
12 note, but it's up to you.

13 And the procedure I use is that I give those notes to  
14 the lawyers and eventually they will -- unless it's  
15 inadmissible, they will try to work in that question with some  
16 witness somewhere. It may be two days or two weeks from now  
17 before you hear the answer, but in due course the lawyers know  
18 that somebody has asked that question and they will probably  
19 address it.

20 All right. Now, there is one question that was asked  
21 that is best for me to answer, and it's a question that says:

22 "By what means do the lawyers for one company  
23 obtain the private information or documents,  
24 i.e., meeting reports, personal emails,  
25 et cetera, with which they confront witnesses

1                   of the other company?"

2                   That's a very good question. The others were, too,  
3 but this one is one I can answer for you.

4                   We have a trial going right now, but before the trial  
5 starts, the lawsuit, I believe, was filed about 18 months ago,  
6 somewhere in there; 16, 18. And the first phase of any lawsuit  
7 is called discovery. That's what the lawyers call it.  
Discovery. And so in addition to having investigators if they  
9 want, any company can -- or any party is entitled to ask the  
10 other side to produce certain records for examination and  
11 copying.

12                  So one side could say, make a formal request: Let us  
13 see all of your emails that mention the word "Android" or  
14 mention the word "Java" or within a time frame perhaps. And so  
15 then the responding company will go back and do a search  
16 through their big servers, pull out those records and let the  
17 other side see them.

18                  And it's not limited to electronic material. It  
19 could be: Let us see all of your hard copy memos to file or  
20 your handwritten notes or let us see the desk file that Mr.  
21 Ellison kept in his desk drawer about this project. And the  
22 parties are entitled to see the originals by the way, if they  
23 want, not just the copies.

24                  And it works both ways. Both sides can do this. And  
25 I promise you both sides have exercised their right vigorously

1 in this case to see the other side's materials.

2 So lawyers come prepared because they have seen the  
3 internal workings of the other side's thought process, the  
4 communications back and forth. And all of this is perfectly  
5 proper. There is nothing sneaky at all about it. It's  
6 perfectly proper to do this in litigation. It works both ways.  
7 Both sides have had this opportunity and you will see both  
8 sides using what looks like confidential material inside the  
9 company here in this public trial.

10 So that is the way in which we come to court so  
11 that -- in the old days you, the jury, would sit here for  
12 months while we subpoenaed in the people and we did this, this  
13 hard discovery just here in front of you. It would take us  
14 months to do that, but in more modern times we spare the jury  
15 all of that and the lawyers and the judge get it all done  
16 before the trial starts so that all of this has been rehearsed  
17 before -- not rehearsed, but the records have been reviewed,  
18 the records have been examined carefully and both sides have  
19 had these opportunities.

20 I'm going to guess millions of pages -- I don't know,  
21 you lawyers can correct me. Over a million pages of records  
22 have probably been produced and reviewed by the two sides.

23 Is that a fair statement?

24 **MR. PURCELL:** I think so, your Honor.

25 **THE COURT:** So you can see the -- you will just see

1 the tip of the iceberg of the records that have actually been  
2 reviewed in this case. You won't see a million. You will see  
3 a couple hundred probably, maybe a few more than that. But  
4 it's definitely the tip of the iceberg because as in all  
5 things, most of them are not that informative, the records, but  
6 there are some that both sides think will help you, the jury,  
7 decide the case.

8 So that's the answer to the question.

9 All right. Now, you will remember that yesterday  
10 when we broke we had Mr. Reinhold on the stand.

11 Mr. Reinhold -- am I saying that right, Reinhold?

12 **THE WITNESS:** Yes.

13 **THE COURT:** (Continuing) -- is still on direct  
14 examination, and he has about 20 minutes to go Mr. Jacobs told  
15 me yesterday. Right?

16 **MR. JACOBS:** Yes, your Honor.

17 **THE COURT:** So please pick it up right there, and  
18 continue right on.

19 **MARK B. REINHOLD,**  
20 called as a witness for the Plaintiff herein, having been  
21 previously sworn, resumed the stand and testified further as  
22 follows:

23 **DIRECT EXAMINATION RESUMED**

24 **BY MR. JACOBS:**

25 **Q.** Dr. Reinhold, we left off yesterday talking about one the

1 Application Programming Interface packages that you have worked  
2 on, the java.nio.channels. And I had asked you whether  
3 java.nio.channels is anywhere depicted, described, mentioned in  
4 the Java language specification.

5 Do you know the answer to that question?

6 **A.** Yes, I do.

7 **Q.** What is the answer?

8 **A.** It appears nowhere in the Java language specification.

9 **Q.** Where is java.nio.channels documented?

10 **A.** It is documented in the Java platform API specification.

11 **Q.** And just to bring us all back, when you say it's  
12 documented in the Java language API specification, what are you  
13 refusing to?

14 **A.** Sorry. It was the Java platform API specification.

15 **Q.** I'm sorry. I said that wrong, didn't I?

16 **A.** I'm referring to the 11,000 pages we discussed yesterday.

17 **Q.** Now, how would one find those pages if one wanted to find  
18 them?

19 **A.** One can find them on the internet, on the worldwide web at  
20 a well-known location.

21 **Q.** Are there copyright notices appended to that  
22 documentation?

23 **A.** There is a link -- there is a copyright notice at the  
24 bottom of every web page and that links to the full legal  
25 notice for that collection of pages.

1 Q. Have you looked at registrations of copyrights in this  
2 case?

3 A. I have looked at registrations, yes.

4 Q. Was it Sun's practice to register copyrights on the Java  
5 platform with the copyright office?

6 A. Yes, it was.

7 Q. I have got some exhibits in front of you I would like to  
8 move through as quickly and efficiently as possible. I have a  
9 list of those exhibits for you because some of those aren't  
10 marked.

11                   **MR. JACOBS:** Your Honor, I have shown this to counsel  
12 for Google.

13                   (Whereupon, document was tendered  
14                   to the witness.)

15 **BY MR. JACOBS:**

16 Q. What is 610.2?

17 A. 610.2, this is a DVD containing an electronic copy of the  
18 Java 5 -- the JDK documentation. It includes the API  
19 specification for Java 5.

20                   **MR. JACOBS:** Move into evidence.

21                   **MR. PURCELL:** No objection.

22                   **THE COURT:** Received.

23                   (Trial Exhibit 610.2 received  
24                   in evidence)

1 BY MR. JACOBS:

2 Q. Dr. Reinhold, what is the Java programming language as a  
3 formal -- in as formal a way as you can describe it?

4 A. As formal a -- so the Java programming language, like any  
5 language, really, has two fundamental parts. There's the  
6 syntax for the language, which is essentially the grammar. We  
7 all remember in grade school taking English sentences and  
8 decomposing them down into their separate parts. There is a  
9 grammar just like that for the Java programming language and,  
10 really, for any programming language.

11           The other important part of the language  
12 specification is the semantics -- that is a set of rules -- in  
13 this case expressed in English prose in several hundred pages  
14 which specify, as precisely as possible, what every construct  
15 you can express in the language means.

16           So if there is a piece of the grammar, for example,  
17 that has a command for a conditional test, if something is true  
18 then do something, then the language specification specification,  
19 describes in excruciating detail what that kind of a statement  
20 is supposed to mean.

21 Q. Yesterday we saw the Java Language Specification Third  
22 Edition, Trial Exhibit 984. Is there a place in that language  
23 specification where you can find a complete set of the rules  
24 for the grammar or syntax?

25 A. Yes, there is.

1 Q. Where is that?

2 A. I believe that is Chapter 18, near the back.

3           **THE COURT:** Chapter 18? Where is that book?

4           **MR. JACOBS:** It may be back in the boxes. We have it  
5 on the screen.

6           **THE COURT:** What is that?

7           **THE WITNESS:** These are the API books, sir.

8           **THE COURT:** Where is the one you're talking about  
9 here?

10          **MR. JACOBS:** We could scroll to 18 on the screen.  
11 That would probably be the most efficient.

12          **THE COURT:** Fine. Go ahead.

13          **MR. JACOBS:** If we could go to Page 588?

14          **THE CLERK:** I don't know what exhibit you're on.

15          **MR. JACOBS:** 984.

16          **THE CLERK:** Is it admitted?

17          **MR. JACOBS:** It's already admitted, yes.

18          Can you go to the beginning of Chapter 18?

19          (Document displayed)

20          **THE COURT:** It's Page 588 using the books page  
21 numbers. No, you have to go back a page.

22          We'll get it up on the screen, your Honor, in a  
23 second.

24          **THE COURT:** All right. Continue on with something  
25 else then.

1 BY MR. JACOBS:

2 Q. What role did the classes in the Java application  
3 programming interface play as to the grammar of the Java  
4 programming language?

5 A. The classes and interface in the Java platform API do not  
6 appear in the grammar of the Java programming language.

7 Q. Is there a relationship between any of the classes in the  
8 APIs and the Java language specification?

9 A. There is a relationship in that the language specification  
10 depends upon certain of the API classes and interfaces  
11 existent, but it does not specify them.

12 Q. Let's turn to -- so this is the beginning now of Chapter  
13 18 that I was referring to earlier.

14 (Document displayed)

15 Q. Can you summarize what the introduction to Chapter 18 is  
16 telling us?

17 A. So, Chapter 18 collects all of the syntax rules for the  
18 language. These rules are expressed elsewhere throughout the  
19 book in piece-wise fashion as each part of the rules is  
20 discussed.

21 Chapter 18 collects them all together as a single  
22 reference. It's typically used by developers who are writing a  
23 Java Language Compiler so that they can understand how to write  
24 the code they need to write to parse an incoming Java source  
25 code file.

1 Q. Let's turn now to Section 1.3 of this exhibit, Exhibit  
2 984.

3 (Document displayed)

4 Q. Does Section 1.3 describe the relationship of the Java  
5 language specification to predefined classes and interfaces?

6 A. Yes, it does.

7 Q. What does it -- do you recall what it tells us?

8 A. It tells us that the specification refers to certain  
9 classes and interfaces.

10 (Document displayed)

11 A. Here we go. Shall read this?

12 Q. Don't read the whole text, but summarize what it's  
13 explaining for us.

14 A. Okay. It notes that some classes have a special  
15 relationship with the language, including, for example, Object,  
16 class, Classloader and String.

17 Q. Are any of the classes in the Java Application Programming  
18 Interface actually specified in the Java language  
19 specification?

20 A. No. As the paragraph goes on to explain, the language  
21 definition constrains the behavior of these classes and  
22 interfaces, but this document does not provide a complete  
23 specification for them.

24 Q. And where are you directed to find the complete  
25 specification?

1   **A.**   The reader is referred to other parts of the Java platform  
2   specification for such detailed API specifications.

3 Q. Is any class described in any detail at all in the Java  
4 language specification?

5 A. There is one class, `java.lang.Object`. If you recall, that  
6 is the class that all other classes are grouped under. That is  
7 described in some detail, but it is not enough detail that  
8 anyone would consider a specification.

9 Q. Did you look at the Java language specification to  
10 determine whether it refers to any other classes?

11 | A. Yes, I did.

12 | Q. And what kind of analysis did you do?

13 A. So, I did a scan of the text of the specification and  
14 extracted all of the names of classes and interfaces that were  
15 mentioned in a way that indicates they are part of the  
16 specification rather than being part of an example.

17 Q. Is that -- is the result of your work in Trial Exhibit  
18 1062 and 1063? They are in the folder there.

19 | A. Yes, I have them here.

20                   Exhibit 1062 the list of is classes and interfaces  
21 mentioned in the Java Language Specification, Third Edition,  
22 but not counting examples or commentary.

23 Q. And by "mentioned," what's the significance of being  
24 mentioned?

25 A. So, many of these classes are mentioned by name and there

1 may be a very brief statement to the effect that, well, if a  
2 certain kind of error condition arises, then one of these  
3 classes is used.

4           About half of the classes, if you look at the list of  
5 30 of them, are what we call exception or error classes. They  
6 are just used to report error conditions. Something has gone  
7 wrong. They are used to report diagnostic information. So,  
8 those especially.

9           They are mentioned in the Java language  
10 specification. They are not specified. They are mentioned  
11 only by name. There is no mention of what methods might be in  
12 them, what fields they might have. They could have anything,  
13 as far as the language specification is concerned.

14           **THE COURT:** I'm confused on one thing, though. I see  
15 the document on the screen.

16           And when you say "Java language specification," is  
17 that the same thing as the Java program language?

18           **THE WITNESS:** Yes, sir. That is the Java programming  
19 language specification.

20           **THE COURT:** All right. As opposed to using Java very  
21 broadly, you're focusing in on Java programming language  
22 specification. But it doesn't say "programming language." It  
23 just says "Java language."

24           **THE WITNESS:** Your Honor, that is the title of the  
25 book and the name of the specification, but it is exactly about

1 | the Java programming language.

2                   **THE COURT:** All right. Now I understand. Thank you  
3 for doing that.

4                   **MR. JACOBS:** And, your Honor, here is Exhibit 984.  
5 I had it next to me the what time.

8 THE COURT: This is in evidence, right?

9 || MR. JACOBS: Yes.

10 THE COURT: Okay. Great. Thank you.

11 | BY MR. JACOBS:

12 | Q. So 1063 is classes and interfaces that are mentioned.

13                    Does the fact of mentioning mean that they are  
14 required by the Java programming language specification?

15 **A.** I'm sorry. We're still talking about 1062 I think?

16 Q. I'm sorry. You're correct.

17 | A. Okay. I'm sorry. And the question was?

18 Q. Does the fact that they are mentioned mean that they are  
19 required by the Java programming language specification?

20 **A.** Yes. They are required by the language specification, but  
21 they are not specified as part of it.

22 Q. Then what is -- can you explain what 1063 is?

23 A. So 1063 is a somewhat similar analysis. So, 1062 was  
24 about what is in the language specification document.

25 || For 1063 I inspected the source code of the Java

1 Programming Language Compiler. If you recall that's the tool  
2 that takes a Java program in source code form and compiles it  
3 into the zeros and ones that the Java virtual machine can  
4 understand. I inspected the source code of the compiler and  
5 looked for all mentions of any specific class or interface, and  
6 these 39 are the only mentions.

7 So in a high level sense, the Java Programming  
8 Language Compiler only knows about these 39 classes and  
9 interfaces.

10 **Q.** Now, can you just describe from the name of the classes  
11 and interfaces that is listed here, what package -- what  
12 packages we're talking about?

13 **A.** So most of them are in the package called java.lang, and  
14 the java.lang package is named that way because they are  
15 classes and interfaces related -- closely related to the  
16 language in this special relationship.

17 There is one in the java.io package because it has to  
18 do with input/output.

19 There are four in the Java.utils package. The use of  
20 these four is not required by the specification.

21 **THE COURT:** Wait one second. The jury is not seeing  
22 this because it's not in evidence yet.

23 **MR. JACOBS:** I'm sorry. We move 1062 and 1063 into  
24 evidence, please.

25 **THE COURT:** Any objection?

1                   **MR. PURCELL:** No objection.

2                   **THE COURT:** All right both received.

3                   (Trial Exhibits 1062 and 1063 received in evidence)

4                   **THE COURT:** You're on 1063 now, right?

5                   **THE WITNESS:** Yes, your Honor.

6                   **THE COURT:** So let's show that one to the jury.

7                   (Document displayed)

8                   **THE COURT:** Go ahead.

9       **A.** So, to recap. Many of these classes and interfaces in  
10 1063 are in the java.lang package. One called serializable is  
11 in the java.io package because it has to do with input/output.

12                  There is several in a subpackage of java.lang called  
13 java.lang.annotation. There are three of those.

14                  And, finally, there are four in the java.util  
15 package. Those are four in the bottom.

16                  It turns out -- well, three of those are -- the names  
17 of those of three classes are mentioned in the compiler source  
18 code, but they are never actually used. So that's a bug. But  
19 at any rate these classes, those last four are not required by  
20 the specification. They are simply used in the implementation  
21 of the compiler as a convenience.

22       **Q.** When you say a class is required by not specified by the  
23 Java programming language specification, what do you mean?

24       **A.** I mean the Java programming language specification asserts  
25 the existence of the classes and interfaces in Exhibit 1062,

1 but it describes, really, only one of them in any detail and  
2 that detail is partial.

3 **Q.** And that is which one?

4 **A.** That is java.lang.object.

5 **Q.** Let's take a look at a demonstrative you prepared.

6 "Demonstrative" meaning an illustration of the analysis that  
7 you have done.

8 (Document displayed)

9 **Q.** What does this show?

10 **A.** So what this diagram shows -- well, it's a classic Venn  
11 diagram it's called, it's called. So there are two circles  
12 here. The smaller circle represents the size counted in pages  
13 of the Java language specification, which is that exhibit. The  
14 large circle represents the size in pages of the Java platform  
15 API specification in Version 5. And the little white  
16 associated area represents the overlap between the two, just in  
17 terms of which classes and interfaces are mentioned in the  
18 specification; the language specification, sorry.

19 **Q.** If we drew this to scale such that the Java APIs  
20 represented by the circle on the right represented the APIs in  
21 Version 5 and, correspondingly, the Java language would be the  
22 Version 3 Java language specification, what would happen to  
23 this diagram?

24 **A.** The big circle would be much bigger.

25 **Q.** And what's the scale representation of the intersection of

1 the two circles?

2 **A.** That intersection would be -- is a lot smaller than it's  
3 shown. It was shown a little bigger because otherwise it would  
4 be very hard to see.

5 **Q.** And we're looking, for the record, at Slide 20.

6 **THE COURT:** Can I ask a question? Now, the 37 APIs  
7 that we're most concerned about in this case, are they all  
8 within that little shaded area or are they somewhere else?

9 **THE WITNESS:** Your Honor, most of them are -- the  
10 bulk of them are outside of that little shaded area.

11 **THE COURT:** In the purple area?

12 **THE WITNESS:** Correct, sir.

13 **BY MR. JACOBS:**

14 **Q.** Without any kind of class library at all -- I want you to  
15 assume no class library -- what kind of programming could you  
16 do with the Java programming language itself?

17 **A.** With no class library at all, you could do very little.

18 **Q.** What would be the first increment you would need to put in  
19 to be able to do something meaningful?

20 **A.** To do something meaningful you would need a way to get  
21 results out of the program. As the Java -- take the Java  
22 language just by itself. It can -- it can take some parameters  
23 to a single method from the command line. If you're typing at  
24 the computer, you can type Java, then the name of your program,  
25 and then some words. And those words are passed in to the

1 program as strings. So there is limited input there, but there  
2 is no way for the program to do output. It can compute. It  
3 can do a lot of computation inside, but it can never  
4 communicate the results of that computation.

5 **Q.** Now, changing the assumption a little bit, you are told  
6 you can create a class library, but you can't copy the Java  
7 platform API specification for any APIs.

8 Can you create your own class library to perform the  
9 functions that you just described?

10 **MR. PURCELL:** Objection. It's a hypothetical.

11 **THE COURT:** Sustained.

12 **BY MR. JACOBS:**

13 **Q.** Does the Java programming language give you the capability  
14 to write your own APIs and class libraries?

15 **A.** Yes, it does.

16 **Q.** Does one need to use Sun's class libraries, Oracle's class  
17 libraries, Sun's APIs, Oracle's APIs in order to program in the  
18 Java programming language?

19 **A.** At the very least you need to use the few that are tightly  
20 related to the Java programming language.

21 **THE COURT:** When you say that, you're pointing at  
22 something. What are you pointing at?

23 **THE WITNESS:** Sir, I'm pointing to the Exhibit 1062.

24 **THE COURT:** And so the few are everything on that  
25 page?

1                   **THE WITNESS:** The few are the -- yes, sir, the 61  
2 mentioned on this page.

3                   **THE COURT:** All right. Thank you.

4                   **BY MR. JACOBS:**

5                   **Q.** Do people design their own APIs for Java?

6                   **A.** Yes, sir. They do all the time.

7                   **Q.** And when they are doing that -- well, how are they doing  
8 that if the APIs are required by the Java programming language?

9                   **A.** I'm sorry. I don't understand the question.

10                  If people are designing their own APIs for Java, then  
11 they are building on top of the Java programming language.  
12 They are building typically on top of all of the standard Java  
13 APIs and creating their own APIs for whatever problem they are  
14 trying to solve.

15                  So, for example, there are -- you know, large  
16 financial firms on Wall Street invest heavily in creating their  
17 own internally APIs for their own class libraries to handle  
18 financial trading activities, but those are not -- those are  
19 strictly built on top of all of this Java platform stuff we  
20 have been speaking about.

21                  **Q.** Do you need the 37 API packages and class libraries at  
22 issue, that we told you are at issue in this case in order to  
23 write useful Java programs?

24                  **A.** No.

25                  **Q.** How many APIs and class libraries were in the first

1 release of Java SE in 1996?

2 **A.** The first release contained seven API packages and, if I  
3 remember correctly, 190 classes.

4 **Q.** Could one write useful Java programs with the first  
5 release?

6 **A.** One could write very useful Java programs with the first  
7 release. That first release was extremely popular.

8 **Q.** Why bother with this separate library called APIs and  
9 class libraries? Why not put it all into the programming  
10 language specification?

11 **A.** Well, from an engineering perspective, that wouldn't be  
12 very smart. It would be kind of like taking, say, the Oxford  
13 English dictionary and adding to it every work of English  
14 literature. They are fundamentally different things.

15 The language is the primitive thing on top of which  
16 everything else is built.

17 The class libraries are written in the language. So  
18 in a sense it's kind of weird to talk about something in the  
19 language being part of the language, if you see what I mean.

20 Another consideration is that the language, the Java  
21 programming language is the thing that every developer using  
22 the Java platform must understand.

23 If you look at the Java APIs, which have grown to be  
24 this rich library of prebuilt components, there are many things  
25 in there that some Java developers will never use. And that's

1 perfectly okay, but the Java language is something that  
2 everyone must understand. So we're very cautious about  
3 evolving it conservatively and keeping it small so that it is  
4 understandable.

5 A final consideration I would mention is that the  
6 Java programming language is -- specification is very tightly  
7 related to the Java virtual machine specification. Now, Java  
8 virtual machines are extraordinarily complex pieces of  
9 software. Many companies have invested a lot in developing  
10 high performance Java virtual machines over the years, not just  
11 Sun or Oracle, and because of that investment they are  
12 reluctant to see the virtual machine specification change  
13 dramatically unless there is good reason. So that's another  
14 reason for keeping the programming language small and, also,  
15 evolving it in an slow, measured pace.

16 **Q.** And how does that compare with development of application  
17 program interfaces and class libraries?

18 **A.** The evolution of the Java platform APIs is more like  
19 wildfire. They have grown just incredibly dramatically over  
20 the years.

21 **Q.** How long had Oracle and its predecessor Sun worked on  
22 developing these APIs?

23 **A.** Since 1996, when Java 1.0 came out and, in fact, some of  
24 the work started several years earlier.

25 **Q.** Have the 37 API packages at issue in this lawsuit

1 themselves evolved over that time?

2 **A.** Yes, they have.

3 **Q.** In what way?

4 **A.** It's very common when you design an API, even if you do a  
5 wonderful job, you probably miss something or maybe there is a  
6 flaw in it that just despite -- no matter how many people you  
7 get to review it, no one notices until the API is out and  
8 published and has been used for awhile.

9 So, you know, little bits of missing functionality or  
10 little bugs in an API will typically be fixed in the next major  
11 release. Maybe you're missing functionality. You add a few  
12 methods. Sometimes we add classes. Sometimes we add  
13 interfaces. Occasionally we add an entire package.

14 **Q.** When you joined the Java software development group in  
15 1996, how many engineers did you have?

16 **A.** I believe there were about 40.

17 **Q.** And did the size of that group change over time?

18 **A.** That group grew fairly dramatically in the late 90's.

19 **Q.** And what was it at its peak?

20 **A.** At its peak it was probably around 500.

21 **Q.** And what were most of those people working on?

22 **A.** Most of them were working on the API specifications and  
23 the class libraries. A significant number were also working on  
24 the Java virtual machine.

25 **Q.** How many were working on the language?

*Katherine Powell Sullivan, CSR, CRR, RPR*

*Debra L. Pas, CSR, CRR, RMR*

*Official Reporters - US District Court - 415-794-6659*

1     **A.** Well, the language, since it evolves so slowly, doesn't  
2 require a large team. Sun almost always had one in-the-house  
3 language expert, somebody typically with a PhD in that area, to  
4 maintain the specification and interpret it, together with a  
5 small team of, you know, maybe two to four engineers to  
6 maintain the Java Programming Language Compiler and enhance it  
7 and so forth.

8     **Q.** Is there still development work going on in Java today?

9     **A.** Yes, there is.

10    **Q.** And that's what you're involved in, right?

11    **A.** Yes, it is.

12    **Q.** And what are you working on?

13    **A.** Right now we're working on Java SE 8, which is scheduled  
14 to ship sometime next year.

15    **Q.** And how many engineers at Oracle are working on the  
16 various Java platforms today?

17    **A.** About 800.

18    **Q.** I would like to do a couple more exhibits there.

19               Sir, do you have Exhibit 623 in front of you?

20    **A.** 623, yes.

21    **Q.** What is 623?

22    **A.** 623 is another DVD, and it is the source codes for Version  
23 1.4 of JDK 1.4, which is the reference implementation of the  
24 Java SE 1.4 platform.

25    **Q.** Do you have Exhibit 980 in front of you? That's one of

1 the books.

2 **A.** Yes.

3 **Q.** What is 980?

4 **A.** 980 is the Java Application Programming Interface  
5 Volume 1. The subtitle is Core Packages.

6 **Q.** How about 981?

7 **A.** 981 is the Java Application Programming Interface  
8 Volume 2. Window, Toolkit and Applets.

9 **Q.** When were those volumes produced?

10 **A.** These were produced in 1996.

11 **Q.** Do they have copyright notices on them?

12 **A.** They do have copyright notices.

13 **Q.** And who wrote these books?

14 **A.** These books were written by engineers and technical  
15 writers at Sun Microsystems.

16 **Q.** Do you have 622 in front of you?

17                   **THE COURT:** Are those in evidence yet?

18                   **MR. JACOBS:** I'm sorry. I offer into evidence, your  
19 Honor, the three items that I mentioned.

20                   **MR. PURCELL:** No objection.

21                   **THE COURT:** 980 and 981 are received.

22                   (Trial Exhibits 980 and 981 received in evidence)

23                   **THE COURT:** What was the third one?

24                   **MR. JACOBS:** 623.

25                   **THE COURT:** Any objection?

1                   **MR. PURCELL:** No objection.

2                   **THE COURT:** Received as well.

3                   (Trial Exhibit 623 received  
4                   in evidence)

5                   **BY MR. JACOBS:**

6                   **Q.** Do you have 622 in front of you?

7                   **A.** Yes, I have 622.

8                   **Q.** What is that?

9                   **A.** 622 is a DVD containing source code for Version 1.4 of the  
10 Java SE platform.

11                   I'm sorry. I may have misspoke on 623. 623 is the  
12 one we just did, correct?

13                   **Q.** 623 is the java.nio.

14                   **A.** My apologies. 623 is the source code for Version 5.

15                   **THE COURT:** All right. Is that the correct?

16                   **THE WITNESS:** That is the correct.

17                   **THE COURT:** So do you want to take back anything?

18                   **MR. PURCELL:** Still no objection.

19                   **THE COURT:** Still no objection. Thank you go ahead.

20                   **BY MR. JACOBS:**

21                   **Q.** And what's 622?

22                   **A.** 622 is the DVD and source code for Version 1.4 of the  
23 platform.

24                   **Q.** And what's -- do you have 965 in front of you?

25                   **A.** Yes, here it is.

1 Q. What is 965?

2 A. So, this is a USB thumb drive containing the source code  
3 for the Java development kit Version 1.1.3.

4 MR. JACOBS: I offer all of these into evidence, your  
5 Honor.

6 THE WITNESS: I apologize again. The testimony  
7 is incorrect. This is the source code for Java Standard  
8 Edition 6.

9 THE COURT: What is the correction you were referring  
10 to?

11 THE WITNESS: That was for Exhibit 965.

12 THE COURT: 965, any objection?

13 MR. PURCELL: No objection.

14 THE COURT: Received.

15 (Trial Exhibit 965 received in evidence)

16 THE COURT: 622, any objection?

17 MR. PURCELL: No objection.

18 (Trial Exhibit 622 received in evidence)

19 THE COURT: Was there anything beyond those two?

20 Mr. Jacobs? 622, 965. What else did you want to move in.

21 MR. JACOBS: 622, 623, 965.

22 THE COURT: 623 is already in.

23 MR. JACOBS: Okay. I think we've got it, your Honor.

24 THE COURT: All right. Continue on.

1 BY MR. JACOBS:

2 Q. And then are you familiar as to 623, 623.1 through 623.10?

3 A. Yes.

4 Q. What are those?

5 A. These are specific source code files taken from the source  
6 code for JDK 1.5.

7 Q. So 623.1, do you know what that is?

8 A. 623.1 is the source code for the class called  
9 java.util.Arrays. That's arrays, plural.

10 Q. And 623.2?

11 A. 623.2 is the source code for a class called  
12 sun.security.provider.certpath PolicyNodeImpl. That's one --  
13 that's three words jammed together, capital P, capital M,  
14 capital I.

15 Q. And 623.3?

16 THE COURT: Is this in hard copy form or is it in  
17 some kind of a --

18 THE WITNESS: This is in hard copy form, your Honor.

19 THE COURT: Thank you. Continue on.

20 A. 623.3 is the source code for a class called  
21 sun.security.acl. -- this is not a commonly used class.  
22 Capital Acl, capital Entry, capital Impl. A software developer  
23 would call that AclEntryImpl.

24 Q. And .4, 623.4?

25 A. .4 is the source code for a class called

1 sun.security.AclImpl; capital Acl, capital Impl.

2 Q. And .5.

3 A. .5 is the source code a class called  
4 sun.security.acl.GroupImpl; capital G and capital I.

5 Q. And 623.6?

6 A. 623.6 is the source code for a class called  
7 sun.security.aclOwnerImpl; O-W-N-E-R, Impl, capital I.

8 Q. 623.7?

9 A. 623.7 is source code for a class called sun.security.acl  
10 PermissionImpl; capital P, capital I.

11 Q. 623.8?

12 A. 623.8 is the source code for a class called  
13 sun.security.acl.PrincipalImpl; P-R-I-N-C-I-P-A-L, capital  
14 Impl.

15 Q. And 623.9?

16 A. 623.9 is the source code for a class called  
17 java.security.CodeSource; capital C, capital S.

18 Q. And 623.10?

19 A. 623.10 is the source code for a class called  
20 java.security.cert, C-E-R-T, CollectionCertStoreParameters.  
21 That's four words with a capital at the beginning of each word.

22 MR. JACOBS: All right. Offer all these into  
23 evidence, your Honor.

24 MR. PURCELL: No objection.

25 THE COURT: Received. So I understand your system,

1 when you move in 623, that was separate from 623.1 through .10?

2           **MR. JACOBS:** Separate as an exhibit demarcation, but  
3 623.1 through .10 are portions of 623. That was the logic.

4           **THE COURT:** But 623 was already in evidence.

5           **MR. JACOBS:** Yes. We wanted to have a separate  
6 exhibit in paper form of specific files.

7           **THE COURT:** I see, okay. Thank you.

8           (Trial Exhibits 623.1 through 623.10 received in  
9 evidence)

10 **BY MR. JACOBS:**

11 **Q.** One last thing, Dr. Reinhold. Could we take a look at  
12 Exhibit 623, please?

13 **A.** 623 again?

14 **Q.** Yes. We'll put it up on the screen.

15           (Document displayed)

16 **Q.** And can we have you take a look at Lines 151 to 152.

17           Do you have a printout of 623 in front you have by  
18 any chance?

19 **A.** Yes, I do -- sorry, no. I don't. I have the DVD.

20           (Document displayed)

21           **THE COURT:** Something is on the screen. Is that what  
22 you wanted?

23           **MR. JACOBS:** Yes, it is, your Honor. Thank you.

24 **BY MR. JACOBS:**

25 **Q.** Could you read Lines 151 and 152, please?

1     **A.**    Yes. So, Line 151 says, "Author Mark Reinhold." Line 152  
2    says, "Author JSR-51 Expert Group."

3                 And I should explain the "author" word there is  
4    preceded by an "@" sign. That's just a notation that's  
5    interpreted by the Java Documentation Extractor so it can find  
6    the names of the authors of a class or interface.

7                 **THE COURT:** Is the asterisk the thing that the  
8    Extractor or whatever uses to identify where the plain English  
9    embedded comments are?

10                **THE WITNESS:** Your Honor, the asterisk is part of the  
11    syntax for that. If we could scroll to -- I guess this won't  
12    really scroll.

13                 Could we show the bottom of the previous page,  
14    please?

15                 (Document displayed)

16     **A.**    Well, that's quite long. I'm sorry. Let's scroll forward  
17    one page and then scroll down a bit and I'll show you an  
18    example.

19                 Okay. Stop right there, please.

20                 Lines 185 through 189. So that's an entire  
21    documentation comment. Comments in Java -- the Java  
22    programming language can be written in a couple ways. In this  
23    case they are written using the asterisks and slashes. So on  
24    Line 185, the slash star is interpreted by the Programming  
25    Language Compiler as the beginning of a comment and it ignores

1 everything up to the closing star slash on Line 189.

2 Now, the Documentation Extractor, unlike the  
3 Compiler, it does pay attention to comments. In particular,  
4 it pays attention to comments which begin with a slash and two  
5 stars. So that's on Line 185. So that's a signal to the  
6 Documentation Extractor, this is a piece of English prose, and  
7 perhaps figures and tables and other kinds of text, that needs  
8 to be reflected into the API specification.

9 **Q.** Is there a copyright notice appended to this file?

10 **A.** There is a copyright notice at the very top this file.

11 **Q.** Can we take a look at that, please?

12 (Document displayed)

13 **Q.** And what does this file relate to?

14 **A.** So this file is one of the files for the Java.nio packages  
15 we were discussing earlier. It's a particular class called  
16 Buffer.

17 **Q.** And can you read the copyright notice for us please?

18 **A.** "Copyright 2004. Sun Microsystems, Incorporated. All  
19 rights reserved."

20 **MR. JACOBS:** No further questions.

21 **THE COURT:** Thank you.

22 Let's start on the cross now. We're going to -- I  
23 need to alert the jury that at 9:00 o'clock I need to be on a  
24 short phone call or as soon as they come out and tell me they  
25 are ready. So I apologize.

1           We're going to break not now, but -- normally we  
2 would break at 9:15, but we'll probably break at 9:00 today,  
3 closer to it. So that would be the reason for the  
4 interruption.

5 All right. Cross-examination. Mr. Purcell, the  
6 floor is yours.

## **CROSS EXAMINATION**

8 | BY MR. PURCELL:

9 | Q. Good morning, Dr. Reinhold.

10 A. Good morning.

11 Q. Dr. Reinhold, the Java language is what's known as an  
12 object-oriented programming language, correct?

13 | A. Correct.

14 Q. And object-oriented programming languages have been around  
15 since the 1960's, correct?

16 A. That's true.

17 | Q. Sun didn't invent object-oriented programming, did it?

18 A. No, it did not.

19 Q. Another object-oriented programming language is a language  
20 called C++, correct?

21 | A. Yes.

22 | Q. C++ also uses APIs, correct?

23 | A. Yes.

24 Q. And another object-oriented programming language is a  
25 language called Objective C, correct?

1 A. Yes.

2 Q. Objective C is the language that the Apple iPhone language  
3 is programmed in?

4 A. Why he.

5 Q. Objective C also uses APIs, right?

6 A. Yes.

7 Q. A third object-oriented programming language is a language  
8 called Python, correct?

9 A. Yes, although Python originally wasn't object-oriented.  
10 It eventually evolved to become that way.

11 Q. Python also uses APIs, correct?

12 A. Yes.

13 Q. Sun didn't invent the concept of APIs, did it?

14 A. No.

15 Q. APIs have been around for many, many years before Java was  
16 invented in the mid-1990's, correct?

17 A. Yes.

18 Q. In fact, there are APIs that exist outside the context of  
19 Java that can be used with a variety of programming languages,  
20 correct?

21 A. I don't understand the question.

22 Q. Well, there are things called language independent APIs,  
23 correct?

24 A. There are things called language independent APIs. They  
25 are generally not very good.

1 Q. But they can be used with language -- independent APIs can  
2 be used with a variety of programming languages as opposed to  
3 just one, correct?

4 A. They can.

5 Q. Now, that's not true of the Java APIs, correct? Those can  
6 only be used with the Java language, correct?

7 A. The Java APIs can be used with the Java language. They  
8 can also be used by -- perhaps not all, but many of the other  
9 languages that can run on top of the Java virtual machine and  
10 the Java Platform.

11 Q. Are the Java API's language independent APIs or are they  
12 language dependent?

13 A. The Java APIs are written in the Java programming  
14 language, but many of the alternative languages that have been  
15 written for the Java virtual machine have a facility inside  
16 them that allows them to call out to the Java platform APIs.

17 Q. Now, yesterday you discussed the various elements of the  
18 design of the Java APIs; do you recall that?

19 A. Yes.

20 Q. The Java APIs are organized into packages, correct?

21 A. Yes.

22 Q. And within packages in Java APIs there are classes,  
23 correct?

24 A. There are classes and there are interfaces.

25 Q. Classes and interfaces?

1                   **THE COURT:** While counsel is writing that, this is a  
2 good opportunity to explain that most likely this is not going  
3 to be in evidence, this chart, but it will -- because this is  
4 illustrative, I assume?

5                   **MR. PURCELL:** Correct.

6                   **THE COURT:** And so if you want to take notes about  
7 it, fine, but it won't be in the jury room.

8                   And sometimes the jury thinks that everything they  
9 see in writing is going to be in the jury room. No, it won't  
10 be. Every illustrative thing you see will not be in the jury  
11 room. It's just to illustrate the testimony.

12                  So if you think there is something there you want to  
13 write down and remember, then you need to be sure to write it  
14 down.

15                  All right. Please continue on with your diagram.

16                  **MR. PURCELL:** Thank you, your Honor.

17 **BY MR. PURCELL:**

18 **Q.** And within classes in the API design in the Java language  
19 there are things called Methods, correct?

20 **A.** Yes.

21 **Q.** I think yesterday you also talked about an element of the  
22 Java API design, something called Fields, correct? Is that  
23 another element of the Java API design?

24 **A.** No, it is not.

25 **Q.** All right. What about exceptions, you discussed

1 exceptions yesterday, correct?

2 **A.** Exceptions are a special kinds of classes, yes.

3 **Q.** And exceptions, that's basically an error, correct?

4 **A.** An exception is a class whose members describe an error  
5 condition, yes.

6 **Q.** Okay. And are exceptions part of the design of the Java  
7 APIs?

8 **A.** Exceptions are part of the design, yes.

9 **Q.** Yesterday Oracle's counsel asked you to explain the  
10 distinction between the Java language on the one hand and the  
11 Java APIs on the other hand, correct?

12 **A.** Yes.

13 **Q.** And in explaining that distinction, you referred to the  
14 Third Edition of the Java Language Specification, correct?

15 **A.** Yes.

16 **Q.** And, in particular, Oracle's counsel walked you through  
17 the table of contents in the Java Language Specification; do  
18 you remember that?

19 **A.** Yes. I believe I walked myself through it.

20 **Q.** Sorry. Didn't mean to rob you of that credit.

21 So let's take a look at Trial Exhibit 984, which is  
22 in evidence.

23 **THE COURT:** Here it is. Do you want it? Dawn.

24 (Whereupon exhibit was tendered  
25 to counsel.)

1                   **MR. PURCELL:** Thank you, your Honor.

2 **BY MR. PURCELL:**

3 **Q.** Now, Exhibit 984 is the document we have been discussing,  
4 the third edition of the Java Language Specification?

5 **A.** Yes.

6 **Q.** Now, let's turn to the table of contents, which is -- in  
7 particular, page lower case Roman numeral ten.

8                   **MR. PURCELL:** All right. Could we blow that up?

9                   (Document displayed.)

10 **BY MR. PURCELL:**

11 **Q.** So Chapter 7 of the third edition of the *Java Language*  
12 *Specification* is called "Packages," correct?

13 **A.** Yes.

14 **Q.** And then at the bottom of that same page we see the  
15 beginning of Chapter 8. And Chapter 8 of the *Java Language*  
16 *Specification*, Third Edition is titled "Classes," correct?

17 **A.** Correct.

18                   **MR. PURCELL:** Could we go to the next page, Ben.

19                   If we could blow up the top. That's great.

20                   (Document displayed.)

21 **BY MR. PURCELL:**

22 **Q.** So this shows the contents of Chapter 8 of the *Java*  
23 *Language Specification*, Third Edition, "Classes," correct?

24 **A.** Yes.

25                   **MR. PURCELL:** And if you look at -- if you scroll

1 down, Ben, to 8.4.

2 **BY MR. PURCELL:**

3 Q. 8.4, section 8.4 is titled "Method Declarations," correct?

4 A. Correct.

5 Q. And there's further sessions that specify further aspects  
6 of methods, correct?

7 A. Correct.

8 Q. 8.4.2 is titled "Method Signatures," correct?

9 A. Correct.

10 Q. 8.4.3 is titled "Method Modifiers"?

11 A. Correct.

12 Q. And below that there are other sections that relate to  
13 methods, "Generic Methods," "Method Return Type," "Method  
14 Throws," et cetera, correct?

15 A. Correct.

16                   **MR. PURCELL:** Could we go to the next page.

17                   (Document displayed.)

18 **BY MR. PURCELL:**

19 Q. Then we come to Chapter 9. Chapter 9 is titled  
20 "Interfaces," correct?

21 A. Yes.

22 Q. And the content of that chapter discusses the ways  
23 interfaces are defined, correct?

24 A. Yes.

25 Q. If you look down at 9.4, at the very bottom of the page

1 there's another reference to methods, correct?

2 **A.** Yes.

3 **Q.** "Abstract Method Declarations" is the title of that  
4 section of the *Java Language Specification*, correct?

5 **A.** Correct.

6           **MR. PURCELL:** Could we go to the next page?

7 **BY MR. PURCELL:**

8 **Q.** About halfway down the page there's the beginning of  
9 Chapter 11, and Chapter 11 is titled "Exceptions," correct?

10 **A.** Yes.

11 **Q.** So all the elements of the API design that we've discussed  
12 today, packages, classes, interfaces, methods, and exceptions,  
13 they are discussed over several separate chapters of the Third  
14 Edition of the *Java Language Specification*?

15 **A.** No, they are not.

16 **Q.** Well, what are all these chapters of the *Java*  
17 *Specification Language* doing if they are not discussing  
18 exceptions, packages, classes, interfaces, and methods?

19 **A.** You're asking me if the elements of the API design are  
20 what is being discussed here. They are not. These are  
21 elements of the Java programming language, which is a different  
22 thing.

23 **Q.** Now, the Java APIs, those are the interface that a  
24 programmer writing an application in the Java language uses to  
25 interact with the source code in the class libraries, correct?

1     **A.**     No.

2                 **MR. PURCELL:** Your Honor, I would like to play as a  
3 party admission Dr. Reinhold's deposition at page 18, lines 4  
4 through 9.

5                 **THE COURT:** Go ahead.

6                     (Video deposition clip played in open court; not  
7 reported.)

8                 **MR. JACOBS:** Your Honor, that was not a prior  
9 inconsistent statement.

10                 **MR. PURCELL:** Your Honor, I think it was. And, also,  
11 it's a party admission. It's not being offered solely for  
12 impeachment.

13                 **THE COURT:** Isn't this a party deposition?

14                 **MR. JACOBS:** It is Dr. Reinhold's deposition.

15                 **THE COURT:** Well, Counsel, it's a party deposition,  
16 right?

17                 **MR. PURCELL:** Yes, Your Honor.

18                 **MR. JACOBS:** Yes.

19                 **THE COURT:** So then it can be used for any purpose  
20 under Rule 32. So it doesn't necessarily have to be  
21 impeaching.

22                     Next question.

23     **BY MR. PURCELL:**

24     **Q.**     Dr. Reinhold, to access the underlying source code and  
25 application in the class libraries, an application has to go

1 through the APIs; doesn't it?

2 **A.** I'm sorry. The question is malformed.

3 **Q.** If a programmer who is writing an application in the Java  
4 language wants to access the source code in the Java class  
5 libraries, that programmer has to go through the application  
6 programming interface, correct?

7 **A.** I'm sorry, the libraries do not contain source code. A  
8 Java developer writing an application is not doing anything  
9 having to do with accessing source code.

10 **Q.** Doesn't the application call on the source code in the  
11 class libraries?

12 **A.** Of course not.

13           **MR. PURCELL:** Let's play Dr. Reinhold's deposition at  
14 page 18, lines 10 through 12, as a party admission.

15           (Video deposition clip played in open court; not  
16 reported.)

17 **BY MR. PURCELL:**

18 **Q.** Now, if there were no APIs, the Java programming language  
19 wouldn't be much use; would it?

20 **A.** If there were no APIs at all, the Java programming  
21 language would not be of much use. That's correct.

22 **Q.** If there were APIs, the programmer could write some basic  
23 computations in the language, correct?

24 **A.** Yes. Or even very sophisticated ones, but they wouldn't  
25 be very useful.

1 Q. That's because if there were APIs, a programmer couldn't  
2 write anything that then could communicate with the outside  
3 world, correct?

4 A. That's correct.

5 Q. Without APIs, a Java program couldn't communicate with a  
6 computer monitor so you could actually read the output,  
7 correct?

8 A. That's correct.

9 Q. And without APIs, a Java program couldn't communicate with  
10 a print so you could print out the output, correct?

11 A. That's correct.

12 Q. Without the Java APIs, a programmer could do computation  
13 with numbers but couldn't really do anything else with them,  
14 correct?

15 A. Computations with number strings, the other primitive data  
16 types.

17 Q. That wouldn't be very exciting for a programmer, would it?

18 A. No, it would not.

19 Q. Yesterday you talked a little bit about the Java Community  
20 Process. Do you recall that?

21 A. Yes.

22 Q. And you talked about the development of APIs within the  
23 JCP, correct?

24 A. Yes.

25 Q. You mentioned that each Java API has what's called a

1 specification lead. Do you recall that?

2 **A.** Yes.

3 **Q.** And that's the person who is in charge, broadly speaking,  
4 of developing a particular API?

5 **A.** Yes.

6 **Q.** Now, for some of the Java APIs in the Java platforms the  
7 specification lead is actually somebody other than someone at  
8 Sun and now Oracle, correct?

9 **A.** Correct.

10 **Q.** In fact -- strike that.

11 So that means that some of the development some of  
12 the Java APIs, they are managed outside of Sun or Oracle?

13 **A.** Correct.

14 **Q.** And, in fact, for some Java APIs, Sun and Oracle don't  
15 even own the copyrights in the underlying source code, correct?

16 **A.** That is true.

17 **Q.** Those copyrights are owned by other members of the Java  
18 community, correct?

19 **A.** Correct.

20 **Q.** Now, even as to the Java APIs where Sun owns the  
21 copyrights, those APIs were created not just solely by Sun but  
22 in collaboration with other members of the Java community,  
23 right?

24 **A.** That's right.

25 **Q.** In fact, you discussed yesterday your own experience in

1 the specification lead of the java.nio package, correct?

2 **A.** Yes.

3 **Q.** And that API was created in collaboration with other  
4 members of the Java community?

5 **A.** Yes, it was.

6 **Q.** You described that as an expert group?

7 **A.** Yes.

8 **Q.** The java.nio expert group, that included a representative  
9 from Oracle Corporation prior to Oracle buying Sun, correct?

10 **A.** Yes.

11 **Q.** And it included a representative from IBM, correct?

12 **A.** Yes.

13 **Q.** Now, even -- strike that.

14 You discussed yesterday the 37 Java API packages that  
15 are at issue in this case, correct?

16 **A.** Yes.

17 **Q.** You are familiar with those packages?

18 **A.** Yes.

19 **Q.** As to ten of those 37 Java API packages, they were created  
20 in collaboration with other members of the JCP; weren't they?

21 **A.** I would have to check, but that could be the right number.

22 **Q.** When APIs are created in collaboration with other members  
23 of the JCP, these experts, these outside experts, they are not  
24 paid for their time, correct?

25 **A.** Not by the employer of the spec lead.

*Katherine Powell Sullivan, CSR, CRR, RPR*

*Debra L. Pas, CSR, CRR, RMR*

*Official Reporters - US District Court - 415-794-6659*

1 Q. They get compensation, presumably, from their employer in  
2 the regular course of business?

3 A. I assume so.

4 Q. But they are not specifically compensated for their work  
5 on the JCP, developing these APIs?

6 A. Not by Sun or Oracle, no.

7 Q. In fact, Google was a collaborator on the development of  
8 some of the Java APIs; wasn't it?

9 A. Yes.

10 Q. You mentioned during your direct testimony this morning  
11 the JSR-51 expert group?

12 A. Uh-huh.

13 Q. You participated in that, correct?

14 A. I was the spec lead for JSR-51.

15 Q. And there was a representative of Google on the expert  
16 group in JSR-51; wasn't there?

17 A. You know, I honestly doesn't recall. It was quite a few  
18 years ago.

19 Q. A gentleman named Matt Welsh?

20 A. Matt Welsh was not at Google at that time.

21 Q. All right. In fact, Google was actually the specification  
22 lead on several JSRs developed through the JCP, wasn't it, over  
23 the years?

24 A. I can think of at least one off the top of my head.

25 Q. Dr. Reinhold, roughly 10 to 20 percent of the Java SE API

1 packages were developed by companies other than Sun or Oracle,  
2 correct?

3 **A.** That could be.

4 **Q.** You can't say for sure?

5 **A.** I would have to go inspect the list.

6 **Q.** And roughly 10 to 20 percent of the Java SE source code  
7 implementing those API packages was written by companies other  
8 than Sun or Oracle, correct?

9 **A.** That could be.

10           **MR. PURCELL:** I would like to play as a party  
11 admission Dr. Reinhold's deposition, pages 111, line 23, to  
12 pages 112, line 12.

13           **THE COURT:** Go ahead.

14           (Video deposition clip played in open court; not  
15 reported.)

16           **MR. PURCELL:** Your Honor, if this is a good time to  
17 break, I'm about to move on to a new topic.

18           **THE COURT:** All right. This will be a good time to  
19 break.

20           The jury will please remember the admonition. No  
21 talking about the case. And when you writeup those notes,  
22 don't discuss what you're writing with anyone else. Just write  
23 it out. It's an individual question. Whenever it's time to  
24 deliberate about the case it will be your duty to talk with  
25 each other about the case, but not yet.

1           All right. Thank you. We'll see you back here in a  
2 few minutes.

3           **THE CLERK:** All rise.

4           (Jury out at 8:57 a.m.)

5           **THE COURT:** All right. Be seated, everyone. Any  
6 issues for the Court?

7           **MR. JACOBS:** Your Honor, I see a potential for  
8 confusion in a complex area about a matter of law. And, so, at  
9 some point we may be asking you for an instruction on this.

10          **THE COURT:** What is that?

11          **MR. JACOBS:** Google's questioning may suggest to the  
12 jury a dispute about ownership of the 37 API packages that are  
13 in dispute here. The jury confuses the percentage of or the  
14 packages or classes or whatever that were developed by  
15 third-parties.

16          There is no ownership dispute here. There is no  
17 question that Oracle has the right, as a matter of ownership,  
18 to assert the copyrights at issue here.

19          **MR. PURCELL:** Your Honor, we're not disputing  
20 ownership of the copyrights. We're responding to a request  
21 from the Court regarding the involvement of other members in  
22 the community in the Java Community Process and API  
23 development.

24          **THE COURT:** Well, if you want to propose something,  
25 I'm happy to consider it. That's okay. But I see other -- I,

1 myself, see other ways in which this line of questions is  
2 relevant to things other than ownership. So it's not -- not  
3 just limited to that potential issue.

4 But to cure that possible misconception by the jury,  
5 if you want to propose a short instruction, I would be happy --  
6 and run it by the other side, I'm happy to give it. All right.

7 **MR. JACOBS:** Thank you, Your Honor.

8 **THE COURT:** We'll take a 15-minute break at this  
9 time.

10 (Recess taken from 8:59 to 9:16 a.m.)

11 **THE COURT:** Please be seated. Back to work. Ready  
12 to go?

13 **MR. JACOBS:** Your Honor, we've had a discussion and  
14 may need the Court's help on this instruction. I am concerned  
15 about confusion here because of the participation of others.  
16 So we would propose that the Court instruct the jury that  
17 Google is not disputing Oracle's ownership of the copyrights as  
18 to the 37 API packages at issue in this case.

19 **THE COURT:** Agreed?

20 **MR. VAN NEST:** No, Your Honor. This is no separate  
21 copyright on an API package. As the witness -- maybe we should  
22 conduct this outside.

23 **THE COURT:** I see the problem. This relates to an  
24 issue I was going to raise with counsel, which is: Does the  
25 Copyright Office do an investigation of the design of the APIs?

1 I rather doubt it.

2           **MR. VAN NEST:** No, they do not.

3           **THE COURT:** So I think this is more complicated than  
4 what you're proposing. That, itself, is subject to some  
5 possible confusion. I'm not saying you're wrong at the end of  
6 the day. I'm just saying that I think there's an issue.

7           **MR. JACOBS:** The point --

8           **MR. VAN NEST:** We can --

9           **MR. JACOBS:** The point is, there is no ownership  
10 dispute.

11           **MR. VAN NEST:** We can work with Mr. Jacobs to get  
12 language --

13           **THE COURT:** I'm not going to do this right now,  
14 anyway.

15           **MR. JACOBS:** Thank you, Your Honor.

16           **THE COURT:** All right.

17           (Jury enters at 9:18 a.m.)

18           **THE COURT:** Be seated.

19           Now, my deputy clerk is not exactly here right now.  
20 Oh, here she is. Okay. I was about to say, if you need some  
21 document we may have to wait.

22           We're ready to go. Everyone is here. Please,  
23 proceed.

24           **BY MR. PURCELL:**

25           **Q.** Dr. Reinhold, I would like to turn back to the Java

1 Language Specification, Third Edition, which is Exhibit 984.

2           **MR. PURCELL:** And, in particular, I would like to put  
3 up page 6 of that document. Could we focus on Section 1.3  
4 there in the center of the page.

5           (Document displayed.)

6 **BY MR. PURCELL:**

7 **Q.** Dr. Reinhold, counsel for Oracle showed you this text  
8 during your direct examination, correct?

9 **A.** Yes.

10 **Q.** You discussed it with him?

11 **A.** Yes.

12 **Q.** This paragraph discusses some classes that have a special  
13 relationship with the Java programming language, correct?

14 **A.** Yes.

15 **Q.** Among those are the classes and interfaces in the package  
16 java.lang.reflect?

17 **A.** Yes.

18 **Q.** So thinking about Java.lang.reflect, I would like to  
19 turn now to Trial Exhibit 1062.

20           **THE COURT:** Is that the new document we saw today?

21           **MR. PURCELL:** Correct.

22           (Document displayed.)

23 **BY MR. PURCELL:**

24 **Q.** Dr. Reinhold, this is a document you prepared, correct?

25 **A.** Yes, it is.

1                   **THE COURT:** That doesn't look like -- what's on the  
2 screen doesn't look like 1062, the one that I saw earlier.

3                   **MR. PURCELL:** I believe it is, Your Honor. It  
4 doesn't have the stamp on it, but I believe it is the same  
5 document I've got.

6                   **THE COURT:** So you're working off one that doesn't  
7 have the stamp.

8                   **MR. PURCELL:** Correct. The one on the screen doesn't  
9 have the stamp.

10                  **THE COURT:** Very well. Go ahead. Does the witness  
11 have 1062?

12                  **THE WITNESS:** Yes. I see it right here, sir.

13                  **BY MR. PURCELL:**

14                  **Q.** Again, Dr. Reinhold, this is a document you prepared,  
15 correct?

16                  **A.** Yes, it is.

17                  **Q.** And the title of it is, "Classes and Interfaces Mentioned  
18 in the Java Language Specification, Third Edition."

19                  Do you see that?

20                  **A.** That's correct.

21                  **Q.** Your intention in creating this document was to provide a  
22 complete list of classes and interfaces mentioned in the *Java*  
23 *Language Specification, Third Edition*?

24                  **A.** Yes.

25                  **Q.** Where in this document could the jury find mention of

1 classes in the package java.lang.reflect?

2 **A.** Those classes are mentioned nowhere in this document.

3 **Q.** Right. But if we could go back to page 6 of Exhibit 984,  
4 Section 1.3.

5 (Document displayed.)

6 **Q.** The text of that paragraph says that the classes and  
7 interfaces in package java.lang.reflect, they are fundamental  
8 to the Java programming language, correct?

9 **A.** No. I'm sorry, it does not say they are fundamental to  
10 the Java programming language. It says that they have a  
11 special relationship with the Java programming language.

12 Moreover, there is no mention in the *Java Language*  
13 *Specification*, Third Edition of any specific classes or  
14 interfaces in the java.lang.reflect -- java.lang.reflect  
15 package. Therefore, they did not show up in the exhibit which  
16 I prepared.

17 **Q.** That's why you omitted them from Exhibit 1062, correct?

18 **A.** That is correct.

19 **Q.** Now, during your direct testimony you talked about a  
20 concept called "write once, run anywhere," correct?

21 **A.** Yes.

22 **Q.** Could you just describe write once, run anywhere for me,  
23 briefly.

24 **A.** So, write once, run anywhere is kind of one of the mottos  
25 of the Java platform. It's the idea that a developer can write

1 an application once, produce it for the Java platform, and  
2 distribute it to customers or users, whoever is going to use  
3 it, and that application will run consistently on any  
4 conforming implementation of the relevant version of the Java  
5 platform.

6 **Q.** Okay. And write once, run anywhere, that historically has  
7 been pretty important to Sun, before Oracle bought Sun?

8 **A.** That was very important to Sun. It's very important to  
9 the entire Java community.

10 **Q.** And it's very important to Oracle now, correct?

11 **A.** Yes.

12 **Q.** Now, Oracle maintains four different Java platforms,  
13 correct?

14 **A.** That's true.

15 **Q.** There's Java SE, Java Standard Edition?

16 **A.** Yes.

17 **Q.** And there's Java EE, Java Enterprise Edition?

18 **A.** Correct.

19 **Q.** That's used on larger devices, like servers?

20 **A.** Yes.

21 **Q.** There's Java ME, which is Java Micro Edition, correct?

22 **A.** Yes.

23 **Q.** That's used on mobile devices such as mobile phones?

24 **A.** Among others.

25 **Q.** And then there's Java Card, correct?

1     **A.**    Correct.

2     **Q.**    And Java Card is used on very small devices like smart  
3 credit cards, correct?

4     **A.**    Correct.

5     **Q.**    A Java ME and Java SE use different versions of the Java  
6 programming language; don't they?

7     **A.**    That's generally true. The version of the language used  
8 in Java ME lags behind a little bit.

9     **Q.**    It lags behind the more complex version of the language  
10 that's used with Java SE, correct?

11    **A.**    Correct.

12    **Q.**    And there are differences between the APIs that are  
13 available in Java SE versus just Java ME, correct?

14    **A.**    Yes.

15    **Q.**    In fact, there are differences in the APIs that are  
16 available across all four of the different Java platforms,  
17 correct?

18    **A.**    Of course, there are.

19    **Q.**    Java ME has a smaller number of APIs than Java SE,  
20 correct?

21    **A.**    Yes.

22    **Q.**    And when a developer is writing Java code, a Java  
23 application, he or she needs to know whether he or she is  
24 writing for Java ME or Java SE, in order to know which APIs  
25 are -- will be available, correct?

1     **A.**    Correct.

2     **Q.**    And that means, in general, that an application written  
3    for a Java SE machine may not work on a Java ME machine because  
4    it may call on APIs that aren't available on Java ME, correct?

5     **A.**    That's correct.

6     **Q.**    Now, let's focus on just the Java ME platform. Within the  
7    Java ME platform there are different what's called  
8    configurations, correct?

9     **A.**    Yes.

10    **Q.**    And configurations, that's sort of a basic layer of APIs  
11    and class libraries that provides a certain functionality,  
12    correct?

13    **A.**    Yes.

14    **Q.**    Is that a good layperson's definition?

15    **A.**    Sure.

16    **Q.**    I'm trying my best.

17               And, initially, when the Java ME platform was  
18    developed, the first configuration that Sun introduced was  
19    something called the Connected Limited Device Configuration; is  
20    that correct?

21    **A.**    Whether it was the first, exactly, I don't recall. I  
22    didn't work on it myself. But that was definitely an early  
23    one.

24    **Q.**    And that was abbreviated to CLDC, correct?

25    **A.**    Yes.

1 Q. CLDC really provided the minimum number of class libraries  
2 and APIs that were required to operate a Java virtual machine,  
3 correct?

4 A. It was a very small number.

5 Q. Subsequent to CLDC, there was a new configuration  
6 introduced on the Java ME platform, and that was called the  
7 Connected Device Configuration or CDC, right?

8 A. Yes.

9 Q. And that didn't have the "L" in it because it wasn't quite  
10 so limited as CLDC, right?

11 A. Correct.

12 Q. It had more APIs?

13 A. Yes.

14 Q. More functionality?

15 A. Yes.

16 Q. So an application that was written for a CDC-running  
17 device wouldn't necessarily work on a CLDC device because there  
18 are more APIs in the CDC configuration, correct?

19 A. It might. It might not.

20 Q. Now, in addition to the platforms and then the  
21 configurations, there's also something called profiles that run  
22 on top of configuration, correct?

23 A. I think it's something like that. I haven't studied the  
24 ME space closely recently.

25 Q. Profiles, though, are an additional layer that provides

1 further APIs and additional functionality above what's  
2 available on just the configuration level, correct?

3 **A.** Yes.

4 **Q.** And, again, on the Java ME platform, the first profile  
5 that was introduced to run on top of the CLDC configuration was  
6 something called MIDP, the Mobile Information Device Profile;  
7 is that right?

8 **A.** I don't recall whether it was first, but that is  
9 definitely one of the ME profiles.

10 **Q.** And MIDP provided additional APIs and additional  
11 functionality that wasn't available in just the base  
12 configuration, correct?

13                   **MR. JACOBS:** Your Honor, scope, foundation.

14                   **THE COURT:** Overruled.

15                   **THE WITNESS:** Yes.

16 **BY MR. PURCELL:**

17 **Q.** And subsequent to MIDP, there was another profile that was  
18 introduced to run on top of the configurations, and that was  
19 called MIDP2, right?

20 **A.** Yes.

21 **Q.** And MIDP2 had yet more APIs that weren't available on  
22 MIDP, correct?

23 **A.** Yes.

24 **Q.** So if there is a device that was a MIDP device, it  
25 wouldn't necessarily be able to run a MIDP2 application,

1 correct?

2 **A.** It might. It might not.

3 **Q.** And subsequent to MIDP2, then there was an MIDP3  
4 introduced, right?

5 **A.** I forget whether it was MIDP3 or MSA, but yes.

6 **Q.** There was a subsequent profile to MIDP2?

7 **A.** Yes.

8 **Q.** Whatever its name might be. And that subsequent profile  
9 to MIDP2 had yet more APIs that weren't available on the  
10 previous MIDP2?

11 **A.** Yes.

12           **THE COURT:** May I? Just remind us, this MIDP, is  
13 this about Standard Edition? What edition are we talking  
14 about?

15           **THE WITNESS:** He's asking about Micro Edition, Your  
16 Honor.

17           **THE COURT:** These -- are these profiles, is that it,  
18 Ps?

19           **THE WITNESS:** Yes, sir.

20           **THE COURT:** That all relates to the Micro Edition?

21           **THE WITNESS:** Yes.

22           **THE COURT:** Thank you. Go ahead.

23 **BY MR. PURCELL:**

24 **Q.** So all these different configurations and different  
25 profiles we've been talking about, the CDC and CLDC

1 configurations, the MIDP, MIDP2, MIDP3 profiles, these are all  
2 within the Java ME platform, correct?

3 **A.** That's correct.

4 **Q.** And Java ME is just one of four platforms that's available  
5 on the Java system, along with EE, SE, and Java Card correct?

6 **A.** Yes.

7 **Q.** So with all these different platforms, configurations, and  
8 profiles, each with different sets of APIs, how can there be  
9 write once, run anywhere?

10 **A.** Write once, run anywhere was never a promise that if you  
11 wrote code for one Java platform that it would  
12 automatically/magically work on another.

13           The write once, run anywhere promise is relative to a  
14 specific one of the Java platforms. If you write an  
15 application that uses Java SE 5, then you can run it on Sun's  
16 implementation, on Oracle's implementation, on IBM's  
17 implementation, and on others.

18           Will that same code run on a particular configuration  
19 of Java ME? Well, it depends. It might. It might not. It  
20 depends which APIs it uses.

21 **Q.** Now, you've familiar with a Sun initiative called One  
22 Java, correct?

23 **A.** Yes, I am.

24 **Q.** And Sun began the One Java initiative sometime in 2008; is  
25 that right?

1 A. Yes.

2 Q. Sun started One Java before Oracle had agreed to purchase  
3 Sun, correct?

4 A. Yes.

5 Q. And One Java was a project that was designed to unify the  
6 Java SE and the higher ends of the Java ME platforms, correct?

7 A. That's correct.

8 Q. And by "higher end" Java ME platforms, I mean the Java ME  
9 that would enable the functionality on a smart phone, correct?

10 A. Yes.

11 Q. Now, the eventual goal of One Java would be that instead  
12 of two separate platforms, Java SE, higher-end Java ME, there  
13 would be just one platform, correct?

14 A. Yes.

15 Q. It would have a unified set of APIs, correct?

16 A. Yes.

17 Q. And a potential benefit of One Java is that it could  
18 address inconsistencies in performance between the Java SE and  
19 the higher-end Java ME platforms, correct?

20 A. Yes.

21 Q. It could further enable the so-called write once, run  
22 anywhere program, right?

23 A. Yes.

24 Q. Oracle officially bought Sun Microsystems in about  
25 February 2010?

1 | A. Correct.

2 Q. In May or June 2010, Oracle decided to pull the plug on  
3 One Java, correct?

4 | A. That sounds about the right time frame.

5 Q. Oracle decided to pull the plug on One Java because it did  
6 not see that project as consistent with its business goals,  
7 correct?

8 | A. Yes.

9 Q. Oracle decided that directly addressing the smart phone  
10 market in that way, through One Java, it wasn't a very  
11 interesting business idea, correct?

12 A. That was my understanding.

13 || **MR. PURCELL:** No further questions.

14 | THE COURT: Redirect.

## **REDIRECT EXAMINATION**

16 | BY MR. JACOBS:

17 Q. Mr. Purcell asked you a question about the use of APIs to  
18 access the source code in libraries, and you told him his  
19 question was malformed. Why was it malformed?

20 **A.** It is malformed because there is no source code in the  
21 libraries. The libraries consist of the 0s and 1s produced by  
22 the Java programming language compiler from the source code.

23 There is no source code in the libraries. It makes  
24 no sense to talk about developers writing applications that  
25 access source code for the APIs.

1   **Q.**   And when you're talking about "in the libraries," the jury  
2 has seen source code for the libraries?

3   **A.**   Yes.

4   **Q.**   So what's the distinction?

5   **A.**   The source code for the libraries, as I said, is processed  
6 by the compiler to produce the 0s and 1s which wind up in the  
7 libraries. Those 0s and 1s, all those bytes are things that  
8 the Java virtual machine can run directly. But to a human they  
9 make no sense, which is why we have programming languages like  
10 the Java language.

11   **Q.**   And these libraries that you're talking about being in  
12 object code form, being in binary, in 1s and 0s, are these the  
13 same libraries that the jury has seen in source code form?

14   **A.**   Yes.

15   **Q.**   And what's the -- in what situation are they usable, or  
16 why would one look at them in source code form if they have no  
17 role in the actual running of a program?

18   **A.**   Well, a developer might look at the source code in case  
19 they're using -- they're using one of the classes in the  
20 library and they see that they think maybe it's not behaving  
21 according to its API specification.

22                 This occasionally happens. Maybe it's a bug or maybe  
23 the developer is just confused.

24                 So since most of the code is open source now, a  
25 developer can go find the source code, pull it up and try to

1 understand at a deeper level what's going on. And sometimes  
2 they even find a bug in the source code and, hopefully, they  
3 report it and we fix it.

4 **Q.** And then when the program is actually executing on a  
5 machine, is the programmer's code accessing the API class  
6 libraries with the API class libraries being in source code  
7 form?

8 **A.** No, they are not in source code form. They are in object  
9 form, in 0s and 1s, just as the developers' own code, by then,  
10 is in 0s and 1s.

11 **Q.** And is that why Mr. Purcell's question was malformed?

12 **A.** Exactly so.

13 **Q.** Now, Mr. Purcell also asked you about the fact that in the  
14 Java programming language specification, volume 3, the various  
15 elements of the APIs that you enumerated are discussed.

16 **A.** You mean the various elements of the language.

17 **Q.** Well, explain -- sounds like my question was malformed.

18 So, explain, why these items, packages, classes,  
19 interfaces, et cetera, appear in the Java programming language  
20 specification if they are not themselves the definition of a  
21 Java class library API.

22 **A.** These are some of the most fundamental concepts in the  
23 Java programming language. They are the basic building blocks,  
24 just as the English language has basic building blocks. So  
25 they are defined in the *Java Language Specification*.

1   **Q.**   And how do they relate, then, to whether a designer can  
2   create his own structure and organization for APIs when  
3   actually writing APIs?

4   **A.**   These are exactly the tools that an API designer uses.  
5   They are the fundamental concepts, the atoms, if you will, from  
6   which an API designer can build a more complex molecule of a  
7   whole API, and then an implementation.

8   **Q.**   So does the Java programming language specification  
9   constrain a developer's ability independently to create unique  
10   and different API specifications?

11           **MR. PURCELL:** Objection. Leading.

12           **THE COURT:** It is leading, but overruled. Go ahead.

13           **THE WITNESS:** Does it constrain it?

14           A developer's API, all it has to do is follow the  
15   rules of the Java language; otherwise, there is immense  
16   freedom.

17           **MR. JACOBS:** Thank you very much, Dr. Reinhold.

18           **THE COURT:** Anything more?

19           **MR. PURCELL:** No, Your Honor.

20           **THE COURT:** May Dr. Reinhold step down --

21           **MR. PURCELL:** He may --

22           **THE COURT:** -- and be excused?

23           **MR. PURCELL:** We're going to need to recall him in  
24   our case because of the limited scope of direct examination.  
25   Reserve that right.

1                   **THE COURT:** Dr. Reinhold, they will get in touch with  
2 you when you are next needed. You are free to go now. Have a  
3 great day.

4                   (Witness steps down.)

5                   **THE COURT:** Oracle may now call its next witness.  
6 And someone should come up here and clear away the witness  
7 stand.

8                   **MR. JACOBS:** Call Mr. Josh Bloch, Your Honor.

9                   **THE COURT:** Welcome. Stand about right there and  
10 raise your right hand.

11                   JOSHUA BLOCH,  
12 called as a witness for the Plaintiff herein, having been first  
13 duly sworn, was examined and testified as follows:

14                   **THE WITNESS:** I do.

15                   **THE CLERK:** Thank you.

16                   **THE COURT:** Welcome. Have a seat.

17                   **MR. JACOBS:** Just a minute, Your Honor.

18                   **THE COURT:** See the microphone, how it moves? See  
19 mine. You've got to be this close.

20                   **THE WITNESS:** That about right?

21                   **THE COURT:** Point it down more. It's too high.

22                   **THE WITNESS:** Is that about right?

23                   **THE COURT:** That's very good.

24                   All right. Mr. Jacobs, the floor is yours.

1 **DIRECT EXAMINATION**

2 **BY MR. JACOBS:**

3 **Q.** Good morning, Mr. Bloch.

4 **A.** Good morning.

5 **Q.** You worked at Sun Microsystems from August 1996 through  
6 July 2004?

7 **A.** That is correct.

8 **Q.** Your title there was senior staff engineer and -- your  
9 titles included senior staff engineer and distinguished  
10 engineer?

11 **A.** Yes, those were my titles.

12 **Q.** You consider yourself one of the architects of the Java  
13 platform?

14 **A.** I am.

15 **Q.** You wrote a book in 2001, called *Effective Java*?

16 **A.** Yes, I did. The first edition was published in 2001.

17 **Q.** You were hired by Google in what year, sir?

18 **A.** In 2004, in July.

19 **Q.** You occasionally use the title chief Java architect at  
20 Google; correct, sir?

21 **A.** That was a courtesy title.

22 **Q.** But you occasionally use that title at Google; correct,  
23 sir?

24 **A.** Yes. Just as I occasionally use the title core libraries  
25 architect at Sun.

1                   **THE COURT:** Pull the microphone. It's still too far.

2                   **THE WITNESS:** Sorry.

3 **BY MR. JACOBS:**

4 **Q.** In fact, Google employees have referred to you as a Java  
5 guru because of your deep knowledge of Java?

6 **A.** Yes. And, actually, not just Google employees. Many  
7 people have referred to me as that.

8 **Q.** Now, you were actually a member of the Android team from  
9 January to December 2009, correct?

10 **A.** Yeah. I might even have joined in December of '08. I'm  
11 not sure.

12 **Q.** You have extensive experience designing and writing  
13 application programming interfaces; correct, sir?

14 **A.** I do.

15 **Q.** A lot of that experience was gained at Sun; correct, sir?

16 **A.** Not really, because I've been doing it for 30 years. So  
17 some of it was gained at Sun.

18 **Q.** You wrote the APIs for many of the Java class libraries  
19 while you were at Sun?

20 **A.** Many, but not most, yes.

21 **Q.** Your first assignment when you joined Sun included writing  
22 security APIs for the Java platform?

23 **A.** Not really. I assisted a junior engineer by the name of  
24 Benjamin Reneau, in writing those APIs. He had written them,  
25 and I helped him improve them.

1 Q. Your role at Sun included helping to decide what libraries  
2 were needed, to design them and to document them, implement  
3 them, test them, and represent them in internal processes.  
4 Correct?

5 A. Yes. I did all of those things and more.

6 Q. You were a specification lead for multiple Java  
7 specification requests?

8 A. Yeah. I think, about three of them.

9 Q. A JSR is how new functionality is included in certain Java  
10 platforms, correct?

11 A. It is one way that new functionality can be included in  
12 new Java platforms.

13 Q. And one of the JSRs that you were involved in eventually  
14 became known as Java SE, correct?

15 A. No. Those are called umbrella JSRs, and I do not --  
16 certainly, as a Sun employee, I never served on an umbrella  
17 JSR. I served on components -- sorry, served on JSRs that are  
18 components of Java SE, but I don't think I served on the JSRs  
19 that actually were responsible for the entire platform. In  
20 fact, I can tell you all the JSRs I served on.

21 Q. Was one of the JSRs that you were involved in, did it  
22 eventually become known as Java SE, sir?

23 A. No.

24 MR. JACOBS: Your Honor, I would like to read from  
25 page 21, lines 4 through 11.

1                   **THE COURT:** Of the deposition?

2                   **MR. JACOBS:** From Mr. Bloch's deposition, dated  
3 July 8, 2011.

4                   **THE COURT:** All right. Just to remind the jury,  
5 the -- it's normal to read -- sometimes the lawyers prefer to  
6 show the video, but it's perfectly okay just to read without  
7 the video.

8                   Counsel, you must read it exactly the way it is  
9 there, errors and all. Say "question," and then read the  
10 question. Say "answer," and then read the answer.

11                  And if there is a goof-up by the court reporter, you  
12 have to read it with the goof-up, not interlineating. So it  
13 must be an exact reading.

14                  I hear no objection, so please proceed.

15                  **MR. JACOBS:** I think, Your Honor, what I'll do for  
16 clarity is start at line 22 of page 20.

17                  **THE COURT:** Fine. Go ahead. Read, read.

18                  **MR. JACOBS:** "During your" --

19                  **THE COURT:** No.

20                  **MR. JACOBS:** Sorry. Question. Thank you, Your  
21 Honor.

22                  **"QUESTION:** During your time at Sun, working  
23 on libraries, did you get involved in  
24 decisions about the inclusion of additional  
25 libraries that were based on input from

1 participants in the Java Community Process?

2       **"ANSWER:** Yes.

3       **"QUESTION:** Can you describe that process,  
4 please?

5       **"ANSWER:** The entire Java Community Process?

6       **"QUESTION:** No, the choice of libraries  
7 through the JCP.

8       **"ANSWER:** I was specification lead on several  
9 what are recalled JSRs. I'm not even sure  
10 what that stands for, but those are the  
11 efforts through which new functionality is  
12 included in certain Java platforms. And the  
13 one that I'm involved in eventually became  
14 known as Java SE. Was also known variously  
15 as J2SE and the JDK during the times that I  
16 was at Sun."

17       **THE WITNESS:** Am I allowed a response?

18       **THE COURT:** No, not unless counsel asks you a  
19 question.

20                  We now go to the next question.

21       **BY MR. JACOBS:**

22       **Q.** Now, you have been involved in API design while at Google,  
23 as well, correct?

24       **A.** Yes.

25       **Q.** So you've had many years of working in the field of

1 application programming interface design, correct?

2 **A.** Yes, I have.

3 **Q.** And, in fact, you have given presentations on the topic of  
4 application programming interface design for some years?

5 **A.** That is true.

6 **Q.** And you've given that presentation repeatedly, and  
7 including and posting that presentation on the Internet,  
8 correct?

9 **A.** Yes.

10 **Q.** And that presentation is called, "How to Design a Good API  
11 and Why it Matters"; correct, sir?

12 **A.** It is.

13                   **MR. JACOBS:** May I approach, Your Honor?

14                   **THE COURT:** You may.

15 **BY MR. JACOBS:**

16 **Q.** Handing you trial Exhibit 624, sir. Can you identify  
17 trial Exhibit 624 for us?

18 **A.** Thank you. Let me see.

19                   Yes. It says, "Trial Exhibit 624." "How to Design a  
20 Good API and Why it Matters."

21                   And it says, "Joshua Bloch, Google Inc." And this is  
22 the version I gave in 2005, in Antwerp, Belgium.

23                   **MR. JACOBS:** Offer in evidence, Your Honor.

24                   **MR. BABER:** No objection.

25                   **THE COURT:** Thank you. Received. 624 received in

1 evidence. Go ahead.

2 (Trial Exhibit 624 received in evidence.)

3 **BY MR. JACOBS:**

4 **Q.** Now, versions of this presentation were given both at Sun  
5 and Google, correct?

6 **A.** Uhm, at Sun and Google?

7 **Q.** Yes, sir.

8 **A.** I cannot recall giving this presentation at Sun.

9 **Q.** Did you give it while you were at Sun?

10 **A.** Yes, I did.

11 **Q.** And no one at Google has ever told you that that  
12 presentation is inconsistent with Google's views, correct?

13 **A.** No.

14 **Q.** No, I am correct?

15 **A.** No, you are correct.

16 **Q.** Thank you.

17 And you gave that presentation because it contains  
18 extremely important advice to programmers in a field that has  
19 not historically been taught; is that correct, sir?

20 **A.** That is correct.

21 **Q.** And you've refined this presentation over the years and  
22 disseminated it widely, correct?

23 **A.** I did.

24 **Q.** Let's turn to the first page of the presentation, please.

25 And this was the lead-off slide for your

1 presentation, correct?

2 **A.** Yep.

3 **Q.** And, among other things, you explained to your audience  
4 why it was important to even be considering this topic of API  
5 designs, correct?

6 **A.** Yeah. Because, especially back in 2004, 2005, it really  
7 wasn't widely appreciated.

8 **Q.** And, among other things, you said that APIs can be among a  
9 company's greatest assets, correct?

10 **A.** I did.

11 **Q.** But if you do it wrong they can be a huge liability,  
12 correct?

13 **A.** Yes, I said that.

14 **Q.** Bad APIs result in support problems, for example, right?

15 **A.** They do.

16 **Q.** What do you mean by a public API? Is that an API that is  
17 published outside the company that creates it?

18 **A.** Yeah. It's an API that is published so that people  
19 outside of the company can write programs using that API.

20 **Q.** Now, then you turn to the specific members of your  
21 audience in your presentation -- by the way, this was given at  
22 a seminar called JavaPolis, correct?

23 **A.** I wouldn't call it a seminar. Conference.

24 **Q.** It's a conference in Europe, I believe; is that correct?

25 **A.** Yes, Antwerp, Belgium.

1 Q. And you collect -- Java developers are collected at  
2 JavaPolis, and they're your principal audience for this  
3 particular version of the presentation, correct?

4 A. Actually, no, because even in 2005, JavaPolis was more  
5 than just a Java conference. There were programmers of other  
6 languages, as well.

7 So I would say it was just a programmer conference.  
8 It had originally been primarily a Java conference, but by 2005  
9 I believe that there were C# programmers.

10 Q. And you had addressed your audience and said, "Why is API  
11 design important to you?" in the next slide; correct, sir?

12 A. I did, because back then only people who considered  
13 themselves API designers really cared about it. Ordinarily,  
14 programmers didn't really think about APIs all that much.

15 Q. That was really the largest point of your presentation, is  
16 that, API is something everyone should take very seriously,  
17 correct?

18 A. Yes.

19 Q. And, in fact, part of the point here was, look, even if  
20 you think of yourself as just a coder, you really got to  
21 think -- as you're doing your code, you got to think of the  
22 application programming interface for that code, correct?

23 A. Yeah. Even if it's not going to become a public API, even  
24 if it's going to stay within your company, your code will be  
25 better if you think of it as pieces that talk to each other

1 through their APIs.

2 **Q.** Let's turn to the next slide, please.

3 In this slide you outline the kind of measures of  
4 quality for a good API; correct, sir?

5 **A.** Yeah.

6 **Q.** And we're looking at characteristics of a good API in the  
7 JavaPolis presentation; right, sir?

8 **A.** Yeah.

9 **Q.** And you said, among other things, a good API is easy to  
10 learn, right?

11 **A.** Yes.

12 **Q.** Easy to use even without having to reference  
13 documentation, correct?

14 **A.** Yeah, it should be easy to memorize. It should just make  
15 sense.

16 **Q.** And it should be hard to misuse. It should be difficult  
17 to make mistakes using it, right?

18 **A.** Or even impossible, right.

19 **Q.** It should be easy to extend, meaning when you are  
20 designing your API, you should think about how you might want  
21 to enhance that API over time; correct, sir?

22 **A.** Yes.

23 **Q.** And it should be appropriate to the audience, meaning make  
24 sure when you're designing your API you have in mind, as you're  
25 authoring the API, who your reader is going to be, correct?

1     **A.**    Yeah. If you're writing an API that's going to be used by  
2 physicists, you better use terms from physics. If you're  
3 writing an API for bankers, you better use terms from banking.

4     **Q.**    And if you're -- okay. Fine.

5                 So, now, let's turn to the next slide, please. This  
6 was an outline of the rest of your presentation, correct?

7     **A.**    Yes.

8     **Q.**    And by "class" and "method" you're referring to classes  
9 and methods in object-oriented programming?

10    **A.**    Yeah.

11    **Q.**    And Java is one of those kinds of program languages,  
12 correct?

13    **A.**    Yes, it is.

14    **Q.**    And you went through -- in the next slide you said, We'll  
15 talk about the process of API design, correct?

16    **A.**    Uhm --

17    **Q.**    Sorry, I'm looking at --

18    **A.**    The one that I have next says "Gather requirements with a  
19 healthy degree of scepticism." Is that the term?

20    **Q.**    Just before that.

21    **A.**    Yeah, right. So which slide am I on?

22    **Q.**    The one on the screen.

23    **A.**    I don't have that one --

24    **Q.**    You can take a look at your screen.

25    **A.**    Got it. Thank you.

1   **Q.**   And then you went through how a good API designer should  
2 do his -- this authoring task correctly; right, sir?

3   **A.**   Yes, I did.

4   **Q.**   And then in the next slide you said, "Let's go out and  
5 find what people need." Right?

6   **A.**   Uh-huh.

7   **Q.**   And those are often called use cases; correct, sir?

8   **A.**   Yeah. And requirements.

9   **Q.**   And then in the next page you said, "You should start by  
10 writing a short specification," correct?

11   **A.**   Yep, that's what I said.

12   **Q.**   So in this authoring process of writing an API, creating  
13 an API design, start out with a one-page description so you get  
14 it all on one page, and you can look at it and see if it makes  
15 sense, right?

16   **A.**   Yeah. And, more importantly, so you can show it to other  
17 people and they can find out if it makes sense. And if it  
18 doesn't, you can still change the thing easily.

19                 Basically, at that point, you want to be agile. You  
20 want to be able to try the API. If it works, great. If it  
21 doesn't, change it completely without having invested a lot of  
22 time in it yet.

23   **Q.**   Now, I would like you to skip ahead to what you'll see  
24 marked on your version as page 12 of 47. It's called "General  
25 Principles."

1     **A.**   All right. Got it.

2     **Q.**   Now, you're talking about the actual task of structuring  
3       the design of application -- of an application programming  
4       interface, correct?

5     **A.**   Yep.

6     **Q.**   And in the next page you say that, "An API should do one  
7       thing and do it well." Correct?

8     **A.**   Uh-huh.

9     **Q.**   "And the functionality should be easy to explain in the  
10      application programming interface itself." Correct?

11    **A.**   Uhm, yeah.

12           If the application programming interface sort of  
13       starts to get ugly and hard to understand, it's an indication  
14       that you, as the designer of the API, are doing something  
15       wrong, and you should go back to square one and start again.

16    **Q.**   And one of the things you should analyze as you're  
17       deciding what should go in this particular API structure, as  
18       opposed to someplace else, is, can you give it a good name  
19       correct?

20    **A.**   Yes. If the names get long and ugly, that's the API  
21       talking to you and saying, hey, you're screwing up here; try  
22       again.

23    **Q.**   So there's actually a design element in selecting the  
24       structure that you're going to design, and associating with it  
25       a particular name, correct?

1     **A.**    Yeah. Either call it a design element or an engineering  
2 element. It's what we call empiricism. You try it. If it  
3 works, great. If it doesn't, try again.

4     **Q.**    In fact, you said there in your slide, "Good names drive  
5 development," correct?

6     **A.**    Absolutely.

7     **Q.**    And by that you meant that if you can name it well, then  
8 that means you've probably done a good job of selecting not  
9 only the name but the associated meaning of the application  
10 programming interface?

11    **A.**    That isn't exactly what I meant by that comment. What I  
12 meant is, if you get the name right, all of a sudden it will  
13 suggest to you other things that should or shouldn't be in the  
14 API.

15               So, for example, if you have a method that's called  
16 publish, all of a sudden you think, oh, gee, if I'm publishing  
17 this thing, maybe I should be able to subscribe to it, too. Or  
18 maybe I should be able to, you know, say I want a temporary  
19 subscription hold. You've got this metaphor going, and the API  
20 tells you what you should or shouldn't include in the API.

21    **Q.**    Let's turn to the next page, Mr. Bloch, page 14. This  
22 slide says, "APIs should be as small as possible, but no  
23 smaller." Do you see that?

24    **A.**    I do.

25    **Q.**    And this is about the structuring of the -- sort of the

1 size of the API, correct?

2 **A.** Depending on how you define size. I think I go into that  
3 where I say, conceptual weight is more important than bulk.

4 What I really mean by that is, you don't want the  
5 user of an API to have to learn a lot of stuff, because it  
6 would be very difficult for them to remember it and use it  
7 without the documentation. So this is really refining that  
8 idea about keeping the API easy to learn and to use.

9 **Q.** So in organizing your application programming interfaces,  
10 you want to organize them in the way that makes sense to you as  
11 an author of these APIs; make them as small as possible, but no  
12 smaller, correct?

13 **A.** Yes. All I'm saying is that "small" here is sort of a  
14 loaded term. By "small" I'm talking about, as I say on this  
15 slide, conceptual weight. Which is -- which is just how much  
16 stuff is in there?

17 It's not necessarily the number of methods or the  
18 number of classes. It's how much stuff that the users of this  
19 API are going to have to learn is in there.

20 **Q.** If you turn to page 17, now, of the presentation.

21 You went into the question of names in somewhat  
22 greater depth, correct?

23 **A.** Yes.

24 **Q.** And this was -- I remember when we talked earlier,  
25 Mr. Bloch, this -- this phrase struck me as quite powerful in

1 your presentation. "Code should read like prose."

2 Do you see that on your presentation?

3 **A.** Yes.

4 **Q.** What were you driving at?

5 **A.** Well, writing a program is very much a creative process.

6 And if you have good words to use, if the API gives you good

7 words that really mean what it is that you're doing with the

8 API, then once it comes time for you to take that API and write

9 a program with it, the program will read like English text.

10 For example, you know, suppose you have a car class.

11 This is the example I have here. If you have a method called

12 speed, and you have -- then you can say if car.speed is more

13 than twice the speed of light, generate alert, watch out for

14 cops.

15 Now, even though you aren't programmers, you know

16 what that mean.

17 **THE COURT:** "Speed of light." I think you meant

18 speed limit.

19 **THE WITNESS:** Speed limit. I'm sorry.

20 (Laughter)

21 **MR. JACOBS:** You're talking quickly.

22 **THE WITNESS:** You're violating laws if it's faster

23 than the speed of light. Guarantee that.

24 (Laughter)

25 **THE WITNESS:** The point is, if you look at this code

1 example -- can the jury see the code example?

2 **BY MR. JACOBS:**

3 **Q.** On the screen, yes, sir.

4 **A.** All right. It reads almost like English because the API  
5 has provided a good vocabulary to express what it is that it is  
6 providing for the programmers.

7 **Q.** And there's an organizational aspect to this that's  
8 highlighted in your second bullet, correct, sir? That, you  
9 have to be consistent, you have to make sure that the same word  
10 means the same thing, even if it's used across classes. Right?

11 **A.** I would call that a naming aspect, rather than  
12 organizational. I'm saying you use the same name to mean the  
13 same thing.

14 I think I even give an example on a later slide where  
15 I say, don't have one method called "remove" and another called  
16 "delete" if they do the same thing. Call them both "remove."

17 **Q.** On page 20 and 21, you talked about the importance of  
18 authoring an API in a way that keeps in mind that you might  
19 actually affect the performance of the computer, correct?

20 **A.** The performance of the program rather than the performance  
21 of a computer. But, yes, if an API is designed badly, it can  
22 be difficult or impossible to write a fast program with it.

23 **Q.** Now, if we skip ahead a little bit to slide 23 of 47,  
24 we'll see that you turn to the topic of class design.

25 Do you see that?

1     **A.** I do.

2     **Q.** And the next -- page 25 of 47 you talked about  
3 subclassing. Do you see that?

4     **A.** Yes.

5     **Q.** And subclassing is setting up an organizational structure  
6 so you have kind of a chapter and subchapter, right?

7     **A.** Uhm, not really. Because the thing about subclasses is, a  
8 subclasses isn't a part of something in the way that a  
9 subchapter is part of a chapter.

10                 Subclass is a more specific type of something. So,  
11 for instance, you might have military personnel is a class, and  
12 then enlisted men and officers are subclasses. So it isn't  
13 really like a chapter/subchapter-type thing.

14                 Do you understand the distinction?

15     **Q.** Thank you.

16                 If you'd turn to page 29. "Don't violate the  
17 principle of least astonishment." You said there that, "When  
18 you're writing the API, you should keep in mind your audience  
19 and try not to astonish or surprise the user of your API."

20     Correct?

21     **A.** Absolutely.

22     **Q.** Then if we skip ahead, sir, to the end of your  
23 presentation.

24     **A.** Slide number?

25     **Q.** It would be slide 47 to 47.

1     **A.**    Got it.

2     **Q.**    You told your audience, "API design is a noble and  
3 rewarding craft." Correct?

4     **A.**    I did.

5     **Q.**    And, in fact, when -- you designed a set of APIs called a  
6 "collections framework." Correct, sir?

7     **A.**    Yes, I designed those.

8     **Q.**    And you've had people come up to you and tell you that the  
9 collections framework, that set of APIs, changed their life;  
10 correct, sir?

11    **A.**    To the best of my recollection, someone actually said  
12 that.

13    **Q.**    And that was because it was so -- you did a great job on  
14 it, right? That's what he was saying to you.

15    **A.**    That's what he was saying.

16    **Q.**    And those APIs are in the Java SE 5, Standard Edition,  
17 correct?

18    **A.**    Yes, they are.

19    **Q.**    And you wrote those while you were at Sun?

20    **A.**    Uhm, I initially wrote them while I was at Sun, and I  
21 continued adding to them after I left Sun.

22               So, for instance, in Java 6, I have some new  
23 collection APIs with names like Q and Dec, that I wrote as a  
24 Google employee.

25    **Q.**    And under whose spec lead auspices?

1     **A.**    Nominally, Professor Doug Lee. There's something called  
2    JSR-166x. This was sort of a continuing sort of pseudo JSR  
3    that's been going on for the past decade or so, that allows a  
4    group of colleagues ...

5     **Q.**    But that person who came up to you, had obviously left a  
6    vivid impression in your mind, was saying to you: API design  
7    is a noble and rewarding craft. You changed my life with the  
8    quality of your craftsmanship. Correct?

9     **A.**    Yes.

10    **Q.**    And, "Here's what good API design can do," you said in  
11    your presentation, "It can improve a lot of programmers and  
12    users and companies." Correct?

13    **A.**    Yes.

14    **Q.**    And "API design is tough." That's what you were also  
15    saying?

16    **A.**    Yeah. Designing a good API is tough.

17    **Q.**    Like any work of craftsmanship?

18    **A.**    I agree with that.

19    **Q.**    Creating a beautiful painting is tough?

20    **A.**    I'm not sure if that's craftsmanship or art, but I guess  
21    that's a fine distinction.

22    **Q.**    And API design, you said and believe, is a noble and  
23    rewarding craft. Correct, sir?

24    **A.**    Yes, I certainly believe that.

25    **Q.**    And, in fact, people have told you that this presentation

1 was important in shaping their thinking about API design,  
2 correct?

3 **A.** Yes, they have. I was surprised by the impact that it  
4 had.

5 **Q.** There are aesthetic matters in API design; correct, sir?

6 **A.** Yes, there are.

7 **Q.** And it's -- it's not being prissy to think about aesthetic  
8 matters. The aesthetics of an API design are part of this  
9 noble and rewarding craft. Correct?

10 **A.** Yes. Generally, an API that displays good aesthetics will  
11 be easy to use.

12 It's like a car's dashboard. Making it pretty isn't  
13 just about making it nice to look at. The car will be actually  
14 easier to drive if you can see the speedometer.

15 **Q.** More than 20 of the class libraries that you worked on at  
16 Sun have corresponding class libraries implemented in Android,  
17 correct?

18 **A.** I'll take your word for it. I don't know the numbers, but  
19 that sounds entirely possible.

20 **Q.** I need to ask you about rangeCheck, Mr. Bloch.

21 **A.** All right.

22 **Q.** You copied rangeCheck code from Sun code that you wrote  
23 while you were at Sun, and you copied that rangeCheck code  
24 while you were writing Android, correct?

25 **A.** I don't recall.

1                   **MR. JACOBS:** Your Honor, I would like to play a  
2 portion of Mr. Bloch's videotaped deposition from 181, 7 to 14.

3                   **THE COURT:** Go ahead.

4                   (Video deposition clip played in open court; not  
5 reported.)

6 **BY MR. JACOBS:**

7 **Q.** As you wrote Timsort for Google, the Timsort code for  
8 Google, what use did you make of the rangeCheck function in the  
9 Sun code while you were writing the rangeCheck function in the  
10 Google code?

11 **A.** I didn't write Timsort for Google. I wrote it while I was  
12 employed at Google. And that's actually a meaningful  
13 distinction. I wrote --

14 **Q.** When you wrote Timsort while you were at Google, what use  
15 did you make of the rangeCheck function in the Sun code?

16 **A.** I wrote Timsort for inclusion in the OpenJDK, which is  
17 Sun's own implementation of the Java platform.

18                   Actually, I initially wrote Timsort as an experiment.  
19 But when I realized that the thing actually ran fast, my  
20 initial inclination was to say, hey, we should add this to  
21 Java.

22                   And, at that point, the OpenJDK was paramount, in my  
23 mind. And in order to make it easy to put Timsort in the  
24 OpenJDK, it was good engineering to use the same rangeCheck  
25 function.

1 Q. Sir, did you contribute that rangeCheck code to Android?

2 A. Eventually I did, as well as to the OpenJDK.

3 Q. And that Timsort code found its way into Android; correct,  
4 sir?

5 A. That Timsort code did.

6 Q. And it was on the Android version that you could download  
7 from the Android website; correct, sir?

8 A. I certainly believe so. Although, I have not --

9 Q. Isn't it true that you made sure that the signatures  
10 matched from the code that you had written while you were at  
11 Sun into the code that you were working on while you were at  
12 Google?

13 A. That is absolutely true.

14 Q. And so you may have copied that signature from the code  
15 that you worked at Sun while you were at Google, correct?

16 A. Yes.

17 Q. Have you -- I wanted to show you -- I want you to look at  
18 an exhibit and ask you if you recognize it. 794.

19 A. Thanks.

20 (Document displayed.)

21 Q. Do you recognize that exhibit, sir?

22 A. I do.

23 Q. And what does it show?

24 A. It shows two small private utility functions, both named  
25 rangeCheck, that are very, very similar to one another.

1                   **MR. JACOBS:** I offer 794 into evidence, Your Honor.

2                   **MR. BABER:** No objection.

3                   **THE COURT:** Thank you. Received.

4                   (Trial Exhibit 794 received in evidence.)

5 **BY MR. JACOBS:**

6   **Q.** 794 shows the code that's relevant here, that you wrote  
7 at -- while you were at Sun, in arrays.java; correct, sir?

8   **A.** Yes, on the top.

9   **Q.** And in the second box it shows the code for Android that  
10 corresponds to that code; correct, sir?

11   **A.** Correct.

12   **Q.** And that's called Timsort.java, correct?

13   **A.** It is.

14   **Q.** And if you look at it on a line-for-line basis, it's  
15 virtually identical; correct, sir?

16   **A.** Yes, it is.

17   **Q.** Now, you were assigned to the Android project in, you  
18 said, late 2008?

19   **A.** Yeah. It was at the very end of 2008 that I transitioned  
20 from my previous project to Android.

21   **Q.** Was there any discussion with you, at that point, about  
22 whether this would represent some kind of a conflict on account  
23 of your ongoing involvement with Java?

24   **A.** No, not that I recall.

25   **Q.** And how about your prior work for Sun, was there any

*Katherine Powell Sullivan, CSR, CRR, RPR*

*Debra L. Pas, CSR, CRR, RMR*

*Official Reporters - US District Court - 415-794-6659*

1 discussion of any kind of conflict?

2 **A.** Again, I certainly can't recall such a discussion.

3 **Q.** Was there any kind of discussion with you about whether it  
4 was appropriate, in light of any guidelines that had been put  
5 in place for Android's development, whether it was appropriate  
6 for you to work on Android in light of your prior work on Java  
7 while at Sun?

8 **A.** I recall no such conversations.

9 **Q.** Now, you were aware that while you were at Sun, that Sun  
10 regularly and routinely attached copyright notices to the code  
11 that it was writing; correct, sir?

12 **A.** Yes, I was.

13 **Q.** And it attached copyright notices to the documentation it  
14 was publishing, correct, sir?

15 **A.** Yes.

16 **Q.** You were aware Sun was asserting copyright protection over  
17 its programs; correct, sir?

18 **A.** Yes, I was aware of that.

19                   **MR. JACOBS:** No further questions.

20                   **THE COURT:** All right. Let's go to Google.

21                   Reintroduce yourself to the jury.

22                   **MR. BABER:** Your Honor, ladies and gentlemen, I'm  
23 Bruce Baber, one of the counsel for Google.

24                   **THE COURT:** Wonderful. Please keep your voice up  
25 now, or speak more into microphone. The floor is yours. Go

1 ahead.

2 **DIRECT EXAMINATION**

3 **BY MR. BABER:**

4 **Q.** Good morning, Josh.

5 **A.** Good morning, Bruce.

6 **Q.** Could you please give the judge and the jury a brief  
7 summary of your educational background.

8 **A.** I got my bachelor of science in computer science at  
9 Columbia University in 1982. And then I went on to graduate  
10 school at Carnegie Mellon University, in Pittsburgh, where I  
11 spent the next eight years of my life getting a Ph.D in  
12 computer science. And that's, basically, it.

13 **Q.** Did you, in fact, got a Ph.D degree?

14 **A.** I did.

15 **Q.** Is it okay if I call you Dr. Bloch?

16 **A.** Sure.

17 **Q.** Dr. Bloch, the jury has heard you worked at both Sun and  
18 Google during your career. Correct?

19 **A.** That's true.

20 **Q.** Have you worked anywhere else?

21 **A.** Yeah. I worked at a company called Transarc, in  
22 Pittsburgh, Pennsylvania. And I also had some summer  
23 internships at IBM and Bell Labs.

24 **Q.** And what kind of company was Transarc?

25 **A.** Transarc was a software company. We did distributed

*Katherine Powell Sullivan, CSR, CRR, RPR*

*Debra L. Pas, CSR, CRR, RMR*

*Official Reporters - US District Court - 415-794-6659*

1 systems, both what are called transaction systems and file  
2 systems.

3 **Q.** Let's explain a little bit. When you say "distributed  
4 systems," what is a distributed system?

5 **A.** Well, a distributed system -- these days everything is  
6 distributed. Distributed means multiple computers talking to  
7 each other over a network. Back in 1990, when this company  
8 started, that wasn't the norm yet.

9 **Q.** So that might be a number of computers in the same office  
10 that are on the same network?

11 **A.** Yes.

12 **Q.** Could be a bunch of computers that connect over the  
13 Internet?

14 **A.** Yes.

15 **Q.** Those would both be distributed systems?

16 **A.** Yes, they would.

17 **Q.** Does Transarc still exist?

18 **A.** No.

19 **Q.** What happened to Transarc?

20 **A.** Transarc was acquired IBM, I believe, in 1994.

21 **Q.** Did you continue to work for Transarc after IBM bought  
22 Transarc?

23 **A.** For another two years or so I did.

24 **Q.** Then what happened next, in terms of your career?

25 **A.** I saw that there was all this exciting stuff going on, on

1 the west coast, and I joined Sun Microsystems to work on Java.

2 **Q.** During the time when you were at Transarc, what kind of  
3 work did you do?

4 **A.** Uhm, I built modules, components, basically libraries,  
5 both libraries for use outside of Transarc and within Transarc.  
6 And designed them, I designed their APIs, implementations,  
7 tests, specifications, the whole nine yards.

8 **Q.** Now, you just used the phrase, for use inside Transarc and  
9 outside Transarc. And during Mr. Jacobs' examination there was  
10 reference to public APIs.

11 Can you explain how those two things relate to each  
12 other?

13 **A.** Yes. They are really the same thing. A public API is an  
14 API that you write in a company or an organization, an open  
15 source project, and it's used outside of it. It's published,  
16 and anyone can write programs to this API.

17 Whereas, a private API is one that is used inside,  
18 just in the implementation of the public API.

19 **Q.** And during your time at Transarc, did you have any  
20 involvement with APIs?

21 **A.** Yes, I did.

22 **Q.** And what kinds of work did you do with APIs back when you  
23 worked at Transarc?

24 **A.** As I said, I designed them to meet the needs of the users.  
25 Implemented them. I wrote the documentation specifications. I

1 pretty much did it all.

2 **Q.** And that was all prior to the time you went to work at  
3 Sun, correct?

4 **A.** Yes, it was.

5 **Q.** Okay. Now, was your work at Transarc the first exposure  
6 you had had to APIs?

7 **A.** Not by a long shot.

8 **Q.** When were you first exposed to APIs?

9 **A.** The first time I really remember realizing that this was  
10 my calling was the summer of 1982. I had a summer job at Bell  
11 Lab -- no, sorry. Let's try again. That was at IBM, Yorktown  
12 Heights.

13 And it had an experimental, kind of parallelizing  
14 processing IBM 370. It was two computers put together. Back  
15 then, that wasn't the norm. This was kind of a super  
16 experiment.

17 And I was tasked with programming these things. And  
18 I had to do what's called assembly language. It's very low  
19 level and nasty and hard to do.

20 I thought, hey, I could put a nice API on top of this  
21 assembly language and make it easier for everyone at IBM to use  
22 this scarce resource.

23 **Q.** Now, you used the phrase "assembly language." Could you  
24 explain to the jury, what is assembly language?

25 **A.** So computers have what are called instruction sets. And

1 when you're writing a program, you're writing in a nice  
2 high-level language these days. You'll be writing in something  
3 with a name like Java or C or C# or something.

4           But the computer doesn't speak those languages. The  
5 computer itself, the hardware speaks in these binary 1s and 0s.  
6 And assembly language is just one level above the 1s and 0s.  
7 And these days very few humans write assembly language. Humans  
8 write high-level languages, and then compilers or interpreters  
9 turn those high-level languages into these low-level 1s and 0s  
10 for the computer.

11 **Q.** And during that summer internship in 1982, at IBM, did you  
12 write any APIs?

13 **A.** Yes, I did. As I say, I wrote what I thought was a nice  
14 little API for this parallel processing IBM 370. Let everyone  
15 use it.

16 **Q.** In order to get your Ph.D. degree from Carnegie Mellon,  
17 did you have to write a doctoral dissertation?

18 **A.** I did.

19 **Q.** Did your doctoral dissertation have anything to do with  
20 APIs?

21 **A.** Yes, it did.

22 **Q.** And just without getting too technical, could you give the  
23 jury a sense of how your doctoral dissertation related to APIs?

24 **A.** You heard about the collections framework, remember,  
25 something that changed my life. Well, basically, I was

1 implementing the same kind of collections on distributed  
2 systems, where the data was stored on multiple machine machines  
3 so that if one of the machines crashed you wouldn't lose any of  
4 your data. You could keep using the data on the other  
5 machines. That's called fault tolerance.

6 And I had to write APIs to use these distributed  
7 collections, much as the Java collections framework is a set of  
8 APIs to use collections.

9 **Q.** Dr. Bloch, is it fair to say you've been involved with  
10 APIs for your entire 30 years of education and professional  
11 life?

12 **A.** That's a fair statement.

13 **Q.** Dr. Bloch, how did you learn the Java language?

14 **A.** I learned it on an airplane, reading a copy of a book  
15 called *Java In A Nutshell*, by a guy named David Flanagan.

16 **Q.** Do you recall approximately when that was?

17 **A.** Yeah. It was on my way to the job interview for the Java  
18 job at Sun. So that would be in 1996, around, you know, May or  
19 something. I don't know.

20 **Q.** And when you learned the Java language from the *Java In A*  
21 *Nutshell* book, did that include any discussion of APIs?

22 **A.** Yeah. In fact, the entire API set of the platform is  
23 summarized in that book.

24 **Q.** And, now, Mr. Jacobs asked you about some of your  
25 writings. You have written a book, yourself, about the Java

1 language, correct?

2 **A.** I have.

3 **Q.** And it's called *Effective Java*?

4 **A.** Yes.

5 **Q.** Has that book won any prizes?

6 **A.** Actually, it's won a bunch of them. But the one I  
7 remember most clearly is, it won the Jolt Award in 2001.

8 **Q.** And in your book *Effective Java* does it include any  
9 discussion or instruction about the APIs for the Java language?

10 **A.** Yes, it does. It tells people how to use them  
11 effectively. It tells people when there are traps, when there  
12 might be mistakes in those APIs, how to avoid falling into  
13 those traps.

14 **Q.** Now, Mr. Jacobs and you discussed for some period of time  
15 your presentation dec. Do you recall that, about APIs and API  
16 design?

17 **A.** Yeah.

18 **Q.** I would like to back up just a step and make sure the jury  
19 understands what you mean when you say an API.

20 **A.** Okay.

21 **Q.** How would you define an API?

22 **A.** I would say that an API is -- the names, the words, and  
23 the set of rules that the programmer uses to communicate with a  
24 library.

25 **Q.** Okay. Let me break that down just for us.

1                  You say an API is names or words. More specifically,  
2 what do you mean when you say "names or words"? Names for what  
3 or words about what?

4   **A.**   What we call method names, primarily method names and  
5 field names. Because when you're talking to a library, you  
6 tell it to do something. Compute the maximum. That's the  
7 method. Or you look at a field. Like if you are going to  
8 print something out, you look at a field called "system.out"  
9 and print to it.

10   **Q.**   Okay. And you said an API is names or words and a set of  
11 rules. What did you mean by the set of rules?

12   **A.**   It's like a contract. It's what has to be true before you  
13 call the method, and what will be true after the method has  
14 returned. So those are, in essence, rules for using this  
15 method to communicate with a library.

16   **Q.**   And you just referenced a contract. I'm not sure I  
17 followed. An API is like a contract?

18   **A.**   Yeah. Basically, the library promises to do something for  
19 you. Let's say, you know, you pass it two numbers, and it  
20 promises to give you the bigger of those two numbers.

21                  Now, if you break your part of the deal by passing  
22 something that isn't even a number -- let's say you pass in a  
23 list instead of a number -- then there's no way that it can  
24 obey the contract. Because if I pass in a long list of numbers  
25 and the number 7, and say, which of these numbers is higher,

1 there's no good answer to that question.

2 So, in that way, it's like a contract.

3 Q. And when the programmer speaks to the library to try and  
4 get it to do something for her or him, does the programmer have  
5 to speak in very precise language, or general language?

6 A. Very, very precise. If you get anything even a little bit  
7 wrong, if you type a capital letter when the method name has a  
8 lower case letter in Java your program won't run. It won't  
9 even compile.

10 Q. And what if you don't comply with one of the rules for the  
11 API as a programmer, what would happen?

12 A. If you're lucky, it won't compile and you'll know right  
13 away that you've screwed up.

14 Q. Stop. When you say it won't compile, would you explain  
15 what that means.

16 A. Ah, so there's two stages to it. You write the program,  
17 and then you run it through a compiler. You compile it.

18 What that does is turns the program that you've  
19 written into a form that the computer can actually execute.  
20 It's not assembly language, but one step higher than that. So  
21 that's compilation.

22 And the lucky thing is that if you violate these  
23 rules, often you'll know right away you violated them. So you  
24 can fix the problem before it blows up when someone is actually  
25 trying to use the program. If you're not so lucky, then it can

1 blow up when you run it.

2 Q. I apologize, Dr. Bloch. I interrupted you.

3 You said if the programmer doesn't comply exactly  
4 with all the rules, you said several things could happen. The  
5 first was that it wouldn't compile. What are the others?

6 A. So, basically, the other one is that it does compile, but  
7 it doesn't run properly. And, you know, if you're lucky,  
8 you'll know it's not running properly. It will just blow up.

9 If you're really unlucky, it will appear to be  
10 running properly, but be doing the wrong thing, and all of a  
11 sudden someone will get the wrong amount of change at the cash  
12 register, because the program running inside the cash register  
13 is broken.

14 Q. Dr. Bloch, in the presentation Mr. Jacobs asked you about,  
15 there is a slide and a reference to an API being a, quote,  
16 little language. Do you recall that?

17 A. I do.

18 Q. In what respects is an API a little language?

19 A. It's what we were just talking about before. It's a set  
20 of words. Typically, the words are actually verbs. Do this.  
21 Do that. And maybe they involve verb, noun, phrases. You  
22 know, remove the file. Remove the header.

23 So when you're learning an API, it's like learning a  
24 small new language. And if that language makes sense, the API  
25 is easy to learn.

1   **Q.**   In your testimony during Mr. Jacobs' examination, when you  
2   were talking about your dec you made the point that you want  
3   your API to be easy to learn; is that correct?

4   **A.**   Yes.

5   **Q.**   And you also said you want it to be easy to memorize; is  
6   that right?

7   **A.**   Yes.

8   **Q.**   Why would you want someone to memorize an API?

9   **A.**   Because programmers are lazy. When you are writing a  
10 computer program, you don't have to look at the documentation  
11 all the time. You want to just be able to type, type, type  
12 from your head to the computer, and just have it work.

13   **Q.**   During the years when you worked at Sun and you were  
14 working on the Sun Java language APIs, was that a goal, to try  
15 and have programmers memorize the APIs?

16   **A.**   Yes. In fact, it was particularly important back then  
17 because we didn't have IDEs yet.

18   **Q.**   Can you please explain to the jury --

19   **A.**   Sorry. It's an Integrated Development Environment.

20                 And these days those helpers, while you're  
21 programming that if you have forgotten the API, will kind of  
22 help remind you.

23   **Q.**   So during the time when you were at Sun, it was a goal of  
24 your work on APIs for the APIs to be easy to learn?

25   **A.**   Yes.

1 Q. And easy to memorize?

2 A. Yes.

3 Q. And easy to use without ever looking at any documentation?

4 A. Yes.

5 Q. Now, this may sound like a silly question, Dr. Bloch, but  
6 is an API something you can hold in your hand?

7 A. No. It's not. Because it's like a contract. You can  
8 have a written contract, but even if you burn up the written  
9 contract it's still there. You know. Burn your marriage  
10 contract, and you're still married.

11 So it is the rules that allow you to tell the library  
12 to do what you want. And rules are abstract.

13 Q. But are there things relating to an API that you can hold  
14 in your hand?

15 A. Yes.

16 Q. What are they?

17 A. The documentation for the API. If you write -- which you  
18 do when you're developing an API, hopefully -- instructions on  
19 how to use that API, that's concrete.

20 Q. And what is the role or the purpose of that documentation  
21 you just described?

22 A. Uhm, it's teaching/telling people how to use the API,  
23 reference, letting them look at it while they're writing a  
24 program so they can make sure the program does what it's  
25 supposed to do.

1                   And, also, it's a specification. That API should  
2 hopefully be precise enough that it allows other people to do  
3 an independent lead limitation of the same API.

4 **Q.** Dr. Bloch, is an API like a blueprint in any way?

5 **A.** It's not -- not really like a blueprint because a  
6 blueprint tells you how to build something. And you can build  
7 something in many different ways that implements the same API.

8                   The API, as I said, it tells you how to talk to  
9 something rather than how to build it. So I don't think of an  
10 API as being like a blueprint.

11 **Q.** Does an API itself tell you how to build anything?

12 **A.** Not really.

13 **Q.** Okay. Do you believe that an API serves as a blueprint  
14 even for the libraries that house the APIs?

15 **A.** Once again, you know, it's a stretch. It's not really  
16 like a blueprint because of the fact that a blueprint tells you  
17 how to implement something. It basically says use this timber  
18 here, and this length should be this.

19                   API specifications, APIs don't do that. They are  
20 requirements. They tell you what this thing that you're  
21 building has to do. How to talk to it. How to write a program  
22 on top of it. But they don't tell you how to build it.

23 **Q.** In the Java language, Dr. Bloch, what is it that  
24 determines the organization of the code libraries that  
25 implement the APIs?

1     **A.**   The names of the methods basically determine that, because  
2   names in Java, they have three parts. It's called a  
3   fully-qualified name. It consists of the class -- actually,  
4   the package, the class, and then the method or field.

5                 So the whole name of something might be something  
6   like java.lang.math.cos. So it's not math. It's a package.  
7   And then -- sorry java.lang is a package. Math is a class.  
8   And then cos is the cosine function. And so the name  
9   determines the organization.

10   **Q.**   And is this notion of a fully-qualified name something  
11   that's a part of the *Java Language Specification*?

12   **A.**   Yes. I'm certain that term is in there.

13   **Q.**   If you could give us just an example. You just described  
14   in general terms package.class.method would be the format of --  
15   at least a start of part of a fully-qualified name. Correct?

16   **A.**   Uh-huh.

17   **Q.**   Using the example you just gave, what would be the  
18   fully-qualified name under the *Java Language Specification* for  
19   the math function you just described?

20   **A.**   All right. So java.lang is the name of the package.

21   **Q.**   For the fully-qualified name, would there be anything  
22   before the words "Java"?

23   **A.**   No. And it has to be lower case, by the way. Lower case  
24   java period l-a-n-g period. And then we have an upper case  
25   M- -- because the class names are upper case -- a-t-h. And

1 then a period. And then the method name, which I believe is  
2 cos, c-o-s. I don't do a lot of trigonometry.

3 **Q.** So that is the name of the specific method, and that's  
4 what would be considered to be a fully-qualified name under the  
5 language specification (indicating)?

6 **A.** Yes.

7                   **THE COURT:** Where's the API part on all that?

8                   **THE WITNESS:** That is part of the Java SE APIs.

9                   So this is one small element in the API. When you  
10 say the API, I think I know what you mean, as in, yes,  
11 java.lang is the package. You if you talk about -- sometimes  
12 people here in these trials in form we talk about a number,  
13 like 37 APIs. They mean 37 packages. And java.lang would be  
14 one of those packages.

15                  **THE COURT:** Can you do this. Circle the word  
16 package.

17                  (Counsel complies.)

18                  **THE COURT:** Is that what you're saying, that is, one  
19 API would correspond to that package?

20                  **THE WITNESS:** So, honestly, that isn't a terminology  
21 that I heard before this trial. You know, quantizing APIs is  
22 kind of hard.

23                  But in this trial people have been using the number  
24 of APIs to be the number of packages. When they say, oh, yes,  
25 this contains 43 APIs, they really mean 43 packages. So, yes.

1                   **THE COURT:** But your own document that you did in  
2 Holland had the word "API." You've heard of that word before.

3                   **THE WITNESS:** I have. But what I'm saying is, to me,  
4 a package is a part of an API. A class is a part of an API. A  
5 method is a part of an API.

6                   And I don't talk about a number of APIs any more than  
7 I would talk about a number of meat. You know, meat is  
8 measured in pounds.

9                   There's no good metric for APIs, really.

10                  **THE COURT:** Well, look. We are trying to just get  
11 the terminology down. It says package.class.method. Now, is  
12 the package the API?

13                  Help us understand what more you need to put up there  
14 so that we know what API is versus package, or whether they are  
15 the same. So you describe it however you want. We're trying  
16 to understand.

17                  **THE WITNESS:** So think of it as, you know, your city,  
18 your state, and your street. They are all part of your  
19 address. So, you know, the APIs here are the addresses, and  
20 the package is the city. The class is the street. And method  
21 is the house number, if you want. It's not a great analogy,  
22 but you get the idea.

23                  **THE COURT:** Yes, but where is the API part?

24                  **THE WITNESS:** All three of them are parts of it.

25                  **THE COURT:** And could, in the same API, there be the

1 same package but a different class, and yet a different method?

2           **THE WITNESS:** Yes. So, for example, in java.util we  
3 have that collections framework that I wrote. And there's one  
4 class for lists, and another class for sets, which can contain  
5 duplicates. So those are two different classes in the same  
6 package.

7           **THE COURT:** And so it sounds like the API is at the  
8 package level. No? Yes?

9           **THE WITNESS:** You know, every language has its own  
10 way of breaking these things up. But, actually, class is  
11 somewhat more fundamental to Java than package, simply because  
12 the Java -- at runtime, you know, entire classes are loaded at  
13 once.

14           But, you know, as I say, I think the address metaphor  
15 is pretty good. Which is a more important part of your  
16 address, the street or the city?

17           **THE COURT:** So you didn't answer my question.

18           **THE WITNESS:** I tried.

19           **THE COURT:** Either say yes, no, or it's impossible to  
20 answer yes or no.

21           The question was, it sounds like the API is at the  
22 package level.

23           **THE WITNESS:** Yes, but it's also at the class and  
24 method levels.

25           **THE COURT:** All right. Thank you.

1 I'm sorry for the interruption. Please continue.

2 **MR. BABER:** Thank you, Your Honor.

3 **BY MR. BABER:**

4 **Q.** You were talking about, Dr. Bloch, the concept of a  
5 fully-qualified name in the Java language. Once a  
6 fully-qualified name for a specific method has been chosen, how  
7 does that name dictate the organization or where that piece of  
8 code will reside in the libraries?

9 **A.** Well, if it's a public API, then you -- you have to put --  
10 an entire class goes in one file. And the file has to have the  
11 name of the class.

12 So, you know, in the case of our java.lang.math class  
13 there actually is a file called math.java. And at the top of  
14 that file there's a line that says "Package java.lang." And  
15 that means that class, Math, is inside the package java.lang.

16 **Q.** So just to be clear, the class name is java.lang.math,  
17 correct?

18 **A.** Correct. And that -- by the way, a fully-qualified class  
19 name is the class including the package.

20 **Q.** And then once the name for an individual method or some  
21 other element of the APIs has been chosen, does that name then  
22 tell you where you can find that code?

23 **A.** Yes, it does.

24 **Q.** Okay. And does the language specification for the Java  
25 programming language give you any choice as to how to compose a

1 fully-qualified name?

2 **A.** I'm not sure what you mean, sorry, by "compose."

3 **Q.** Does it have rules about what fully-qualified names have  
4 to look like?

5 **A.** Yeah, it has conventions.

6 **Q.** And prior to this lawsuit, Dr. Bloch, have you ever heard  
7 any discussion about the structure, sequence, and organization  
8 of APIs?

9 **A.** Actually, no. This is the first time I've heard that  
10 term.

11 **Q.** Dr. Bloch, how do the APIs and the language relate to each  
12 other (indicating)?

13 **A.** Uhm, well, in a couple of ways. Language lets you write  
14 your own APIs.

15                   **MR. JACOBS:** Objection, Your Honor. So the record is  
16 clear, Mr. Baber is holding up the *Java Language Specification*  
17 to the witness.

18                   **THE WITNESS:** Okay.

19                   **THE COURT:** I'm not sure. I was maybe two sentences  
20 back. So you may have a good objection and I just don't  
21 understand it. He's --

22                   **MR. JACOBS:** Objection, the question was vague. And  
23 on the record he's holding up the *Java Language Specification*.  
24 The witness may be answering two different questions.

25                   **THE COURT:** That's a good point.

1               Okay. Rephrase the question.

2 **BY MR. BABER:**

3 **Q.** Dr. Bloch, you are familiar with the Java programming  
4 language, correct?

5 **A.** Very familiar.

6 **Q.** How do Java language APIs and the Java programming  
7 language relate to each other?

8 **A.** Well, the Java programming language let's you write your  
9 own APIs, and it contains APIs that are part of Java. And, in  
10 fact, in order to write any Java programs at all, you need to  
11 use those APIs.

12 **Q.** Do you need any APIs in order to implement the language  
13 specification?

14 **A.** Yeah. You do. I mean, because it's an object-oriented  
15 language. Right?

16               So, first of all, there's an API for  
17 `java.lang.Object`. You need that one. You can't have an  
18 object-oriented language without objects. And you need  
19 strings. And, in fact, many of these APIs are actually  
20 referenced directly in the *Java Language Specification*.

21 **Q.** Do you know approximately how many APIs or classes within  
22 the APIs are mentioned directly in the *Java Language*  
23 *Specification*?

24 **A.** As it happens, I do. There are 60 classes mentioned  
25 directly in that specification, consisting of 700 -- more than

1 750 methods and fields spread across three classes.

2 **Q.** And you said -- mentioned "directly." What did you mean  
3 by that?

4 **A.** I mean they are in the normative text. They are not in  
5 examples. They are required in order to faithfully implement  
6 that specification.

7 You cannot implement Java as specified in that book  
8 without providing those 60 classes and their 750 methods and  
9 fields spread across three packages.

10 **THE COURT:** What book is that?

11 **MR. BABER:** Your Honor, may I approach?

12 **THE COURT:** Is that the one we had before, 984.

13 **MR. BABER:** Yes, Your Honor Exhibit 984.

14 **THE COURT:** All right. Is that what you're referring  
15 to?

16 **BY MR. BABER:**

17 **Q.** Dr. Bloch, I'm handing you Exhibit 984.

18 **A.** Did I get to keep it?

19 **THE COURT:** Okay. Now that you have the book there,  
20 repeat what you just said about the 900 something.

21 **THE WITNESS:** All right. So what I said was, in this  
22 book you will find that in order to implement the Java language  
23 you actually have to implement 60 classes. They are mentioned  
24 directly.

25 So, you know, for example, if you -- if you look up

1 arrays, you'll find that an array is an object, and it has to  
2 implement serializable -- so those are two classes right there:  
3 Java.lang.Object and java.io.Serializable. If they are not  
4 there, you're not implementing the Java language as per this  
5 book.

6 This book is the defining document for the Java  
7 language. And in order to actually implement it, for starters,  
8 you need the 60 classes that are included by reference in the  
9 definition of the Java programming language.

10 **THE COURT:** How many APIs do those 60 classes go  
11 with?

12 **THE WITNESS:** So it depends on whether -- remember,  
13 how I sort of said city, street, and house number? So, you  
14 know, the 60 is street. And they fall into three packages.  
15 And inside close classes you'll find that there are over 750  
16 public methods and fields.

17 So of the sort of fine-grained APIs you need the 750  
18 methods and fields. Of the medium-grained ones, called  
19 classes, there are 60. Remember when I said classes are sort  
20 of the more fundamental API units? Those are the ones that are  
21 referenced in here directly.

22 But, implicit in those class names is a package name.  
23 You know, so it will say either Object, which means  
24 java.lang.Object, or it will say java.io.Serializable. Which  
25 means the class Serializable in the package java.io.

1           As I said, there are three packages mentioned  
2 directly in this book. They are java.lang, java.io, and  
3 java.lang.Annotation.

4           **THE COURT:** I understand that part. But three is not  
5 37.

6           **THE WITNESS:** No, it isn't.

7           **THE COURT:** So there are other APIs then, I take it,  
8 that are not referenced in that book, that are at issue in this  
9 case. True?

10          **THE WITNESS:** That is correct.

11          **THE COURT:** All right.

12 **BY MR. BABER:**

13 **Q.** Dr. Bloch, with respect to the 60 classes that you just  
14 told us about, that are in the three packages, are there other  
15 parts of the APIs that are needed in order to implement those  
16 60 classes?

17 **A.** Yes. This is what is called API dependencies. When you  
18 have a method, that method you have to pass things in. Those  
19 are called input parameters. So if you're going to, let's say,  
20 take the maximum of two numbers, you pass in two numbers. If  
21 you are going to sort a list, you have to pass in the list.

22           So if you look at these classes and you look at their  
23 methods, you'll see that they take arguments and they return  
24 values. And those have classes. And those classes are not all  
25 among these 60.

1           You know, so if in order to faithfully implement  
2 "java.lang" you also need to pass in parameters that are of  
3 type -- I don't know, print stream. I'm just making this up.  
4 That's not in there. I pulled one out of the air.

5           But if you need another class, then that indirect  
6 dependency is a requirement to implement Java. You can't  
7 implement the language unless you have all of the classes that  
8 are required in this book.

9           And all of the classes that are required by those  
10 classes, and so on, you just have to follow these dependencies  
11 out. And once you're done with that, you'll find that in order  
12 to faithfully implement the language, you must implement 177  
13 classes -- I believe that's the number; it's over 175 --  
14 consisting of over 2,000 public methods and fields spread  
15 across ten packages.

16           And that's required just to implement the language.  
17 That won't necessarily get you the other things people use when  
18 they write Java programs. At a minimum, that's what you need.

19 **Q.** Dr. Bloch, I take it from your testimony you are familiar  
20 with that book that's in front of you, the *Java Language*  
21 *Specification*, which is trial Exhibit 984?

22 **A.** Yeah. I had to study it in great detail when I wrote and  
23 co-authored the three that I wrote and co-authored.

24 **Q.** The version in front of you is actually the third edition  
25 of the *Java Language Specification*; is that right?

1     **A.** Yeah. It covers Java 5.

2     **Q.** What was the earliest edition that you became familiar  
3 with of the Java programming language specification?

4     **A.** So James Gosling, who is the father of Java, actually  
5 personally handed me a pre-release copy of the first version of  
6 this book in, I think, 1997, or late '96.

7     **Q.** And as the language specification has changed from the  
8 first edition to the second edition to the third edition, have  
9 you continued to remain familiar with it?

10    **A.** I have.

11    **Q.** And thinking back to the first edition of the *Java*  
12 *Language Specification*, did that include any discussion of the  
13 API packages, any of the API packages?

14    **A.** Yeah. Actually, in addition to those that were directly  
15 required to implement kind of the formal language, it contained  
16 the full specifications for all of what were the core packages  
17 at the time.

18               At the time, there were only four core packages.  
19 There was `java.lang`; `java.io`, which is input and output;  
20 `java.net`, which is networking; and `java.util`, which is  
21 utilities.

22               So the first edition of this book actually contained  
23 the entire documentation for all of the core libraries, but  
24 there were only four packages at the time.

25    **Q.** Are you familiar today, nowadays, with a separate book

1 that's published regarding those four core libraries?

2 **A.** Yeah, but it's kind of obsolete.

3 **Q.** And why is it obsolete?

4 **A.** Because we have added so many more libraries since then.

5 **Q.** But back in the First Edition of the Java Language  
6 Specification, were the complete specifications for those four  
7 core packages in the language specification?

8 **A.** I believe they were.

9 **Q.** Now, if you didn't have any APIs, Dr. Bloch, could you do  
10 anything with the Java language?

11 **A.** You could waste time with it, and that's pretty much it.  
12 You could -- you could -- let's say you could add the numbers  
13 from one up to a thousand but then you couldn't even print the  
14 result.

15 **Q.** Could you save the results of your program?

16 **A.** No. That's IO.

17 **Q.** Okay. Could you put it on a disk for later?

18 **A.** Couldn't do that.

19 **Q.** Okay. Could you -- can you think of an example, Dr.  
20 Bloch, that is used in teaching programming of a very simple  
21 basic program that computer programmers learn how to write when  
22 they start learning a new language?

23 **A.** The simplest program that you can write in any language is  
24 called the Hello World Program. It just uses that programming  
25 language to print Hello World, and that's how you sort of start

1 off with on new language. You learn how to write a program  
2 that simply prints Hello World in the language.

3 **Q.** And could you do -- could you print a Hello World Program  
4 in Java without any APIs?

5 **A.** Absolutely not. So the Hello World Program has one line  
6 in it and it says system.out.print, quote, Hello World, end  
7 quote.

8           What's really going on there? Well System, that's a  
9 class. It's java.lang.system. So that's one of those 60  
10 classes. And then that -- that actually turns out to be  
11 required in here. And .out, out is a field. Remember, I was  
12 talking about methods and fields before? System.out is a field  
13 in the System Class and its type is, I believe, PrintWriter.  
14 PrintWriter is a class in the Java.io API.

15           And then when you pass in Hello World, that's a  
16 string of java.lang.string. So even a Hello World program  
17 mentions these three classes which are, as I said  
18 java.lang.string, java.lang.system, and java.io.printwriter.

19 **Q.** Dr. Bloch, back to the presentation that Mr. Jacobs asked  
20 you about. I believe you said that writing code using the APIs  
21 is easier if the API gives you good words.

22           Do you recall that testimony?

23 **A.** Yes. Yes, I do.

24 **Q.** What did you mean "if the API gives you good words"?

25 **A.** If they give you words that allow you to express yourself

1 easily, nice words, short words that clearly express your  
2 intent. Easy words to memorize, words that don't mislead you.

3 **Q.** When you say "words" in that context, are you referring to  
4 the name of the API?

5 **A.** Yeah. I'm actually referring to the names of the classes  
6 and the methods. The package names matter a little bit less,  
7 because programmers don't actually type the packages very  
8 often.

9 **Q.** Is there some respect, Dr. Bloch, in which the names of  
10 the APIs are like words that are used in the Java programming  
11 language?

12 **A.** Yes. They are very similar. The idea is that in order to  
13 do anything with a language, you don't want to be confined to  
14 the small set of words that are the so-called key words of the  
15 language. In Java that means like "while" and "do" and "for."  
16 You want to be able to add your own vocabulary specific to what  
17 you want to do with the language.

18 So if you're going to run the language, say, to run a  
19 cash register, then you want to be able to add a word to  
20 express putting money into the till, taking money out of the  
21 till, and those new words are APIs.

22 **Q.** Dr. Bloch, Mr. Jacobs asked you in your testimony that --  
23 when he was asking you questions, whether or not your  
24 presentation about APIs was limited just to Java or whether it  
25 also had relevance to other programming languages. Do you

1 recall that testimony?

2 **A.** Yes.

3 **Q.** All right. We have been discussing for the past few  
4 minutes the relationship between the Java language and its  
5 APIs.

6 I'd like to ask you now: Is that relationship  
7 similar or different from other object-oriented languages in  
8 terms of how those languages relate to APIs for those  
9 languages?

10 **A.** For most modern-oriented languages, like say C-sharp, it's  
11 pretty much identical. So C-sharp will have a language feature  
12 and that language feature will rely on some APIs. So it's the  
13 same.

14 **Q.** Now, Dr. Bloch, one of the things Mr. Jacobs asked you was  
15 about your duties at Sun, and I believe you referenced at least  
16 one part of it was not just creating new APIs, but  
17 reimplementing or doing further work on existing APIs; is that  
18 right?

19 **A.** That is true.

20 **Q.** Okay. And what I'd like to talk for a few minutes about  
21 is not about designing a new API. I want to talk about  
22 situations where you were reimplementing or trying to improve  
23 an existing API?

24 **A.** Okay.

25 **Q.** All right. Now, in order to do that, I want to first ask

1 you: What is a method declaration in the context of an API?

2 **A.** When you're going to write a method, you have to first say  
3 what its name is and what arguments or parameters it takes on  
4 input. That means what it's going to operate on and what it  
5 returns, the type of the thing that it returns. And you're  
6 allowed to name the arguments.

7 All of those things together constitute the  
8 declaration. And so, for example, if we're going to have a  
9 method to compute the maximum of two integers -- I know that's  
10 a good simple example -- so you can say "Public." And that  
11 means anyone can use this method. It's something that's, you  
12 know, going out there for the world to use.

13 Static. Int, it means I'm going to return an integer  
14 to you.

15 And then max, lower case m-a-x. That's the name of  
16 the method. It's called max. The fully qualified name I can't  
17 tell you without knowing the class and package that this method  
18 is in.

19 So you say Public static int max open paren, and then  
20 you give the type of the first argument, which is also an  
21 integer. So we say int. And let's call it arg1 because it's  
22 the first argument. Static --

23 **Q.** You're saying number one or --

24 **A.** Use numeral one, yes. A-r-g-1.

25 No, no, no. No, that's not allowed. Bzzt. Compiler

1 error.

2 (Laughter.)

3 **A.** Lower case r. A-r-g-1 -- that's all lower case -- comma  
4 int arg2 close paren. And that's the declaration right there.  
5 So that tells you that in order to compute the maximum, you  
6 say -- now, let's assume that it's in a class called Math. In  
7 a package called java.lang because, in fact, there is a max  
8 method in java.lang.math.

9           **THE COURT:** Wait. Does it -- the word "Public," is  
10 that preceded by a word like "declare" or --

11           **THE WITNESS:** No. It's implicit just because of  
12 where it is in the document. Just as in a dictionary, it  
13 doesn't say "definition" before every definition.

14           **THE COURT:** Where would that line appear in the  
15 printout?

16           **THE WITNESS:** Either after the first open bracket at  
17 the top of the file or after the last declaration. The  
18 declaration and the code that implements the declaration, they  
19 just come one after another in the file. And that, by the way,  
20 is defined in this book, the syntax of the language.

21           **THE COURT:** Let's say you've got it written out like  
22 the way you have it there. Do you need any other lines of code  
23 underneath that in order to make it work or is that enough  
24 right there to --

25           **THE WITNESS:** Oh, of course you need lines to make it

1 work. That is just the API there.

2 The API is not -- it's implementation. That tells --  
3 that's the language. That tells you what to do in order to  
4 compute the maximum, but it doesn't tell the computer how to do  
5 it.

6 If we want to tell the computer how to do it, we have  
7 to write some code. We have to write something like if arg1 is  
8 greater than arg2, then return arg1, otherwise return arg2.  
9 You can see how that will always return the higher of the two  
10 things. That's the code. And without that code the program  
11 wouldn't have any idea how to compute the maximum.

12 Computers are very literal. They only do what you  
13 tell them.

14 But that code is not part of the API. It's part of  
15 the library. It's part of the implementation of that API.

16 **THE COURT:** Well, I know the jury would like to  
17 understand this part, too, so tell counsel what to write  
18 underneath there to illustrate.

19 **THE WITNESS:** Can I just go write it?

20 **THE COURT:** You can. Well --

21 **THE WITNESS:** I can tell him. I'll tell him.

22 **THE COURT:** You can go down there and write it, but  
23 use --

24 **THE WITNESS:** A different color?

25 **THE COURT:** Use something that is a better magic

1 marker. I have these.

2 (Whereupon, markers were tendered

3 to the witness.)

4 (Witness steps down.)

5 **THE COURT:** If you think a new page is better to  
6 start --

7 **THE WITNESS:** Yes, I do. Thank you very much.

8 **MR. BABER:** Shall we move it up?

9 **THE COURT:** Well, all right. Maybe about five feet  
10 closer. But I want to see it at well. Turn it so that the  
11 jury can see it.

12 All right. That's good.

13 **MR. BABER:** May I stand over there, your Honor?

14 **THE COURT:** You can stand over there.

15 All right. Now, you remember your back is going to  
16 block the jury so you need to move around every now and then,  
17 but go slowly through what it is you're trying to explain.

18 **THE WITNESS:** I will do that.

19 **A.** So at the very top of the file it's going to say Package  
20 java.math semicolon. What that means is the classes, the  
21 Public classes that are declared in this file are part of the  
22 package java.math or the API java.math, if you will. That's  
23 the way the term has been used in the trial so far.

24 Then we have to declare the class. The class, as we  
25 said, was called Math. It's a Public class. So we say Public

1 Class Math. And that's the name of the class. And the name  
2 after the fully qualified name of that class is then Java --  
3 sorry. This says java.math and that was a mistake. It's  
4 java.lang. I apologize.

5 So the package is called java.lang. The class is  
6 called Math. And then we have to declare the method. So let's  
7 do it.

8 We say Public static int. That means -- Public means  
9 this method is part of the Public API. Static, don't worry  
10 about. It's too technical to get into here, unless you really  
11 want to and then just ask me and I'll do it. Int means returns  
12 an integer that is a whole number one, two or three; not  
13 one-and-a-half or two-and-a-half. And then we have the name of  
14 the method. We want to call it max. And then you put an open  
15 paren to say now I'm going to tell you what arguments we have  
16 to pass in to the method.

17 So we'll say int arg1. And then the first thing I  
18 have to pass in is an integer. And int arg2. Close paren.  
19 That's it for the declaration. That's the API.

20 But then you have a bracket. And down here a closed  
21 bracket. Between them you have to put some actual Java code to  
22 tell the machine how to calculate that maximum. That's not  
23 part of the API. That is part of the implementation.

24 Should I put an implementation that works? It will  
25 just take me 10 seconds.

1                   **THE COURT:** Go ahead.

2       **A.** So I can say if arg1, the first argument, is greater than  
3 arg2, the second argument, then return arg1 else return arg2.

4                   And now think about what happens if I pass in, let's  
5 say, one as arg1 and 13 as arg2. I say if arg1, that's one, is  
6 greater than 13 -- is it? No. So I go down here to the else.  
7 Return arg2. And what was arg2? 13.

8                   So it has correctly calculated the maximum of its two  
9 input arguments. And that's a pretty simple computer program  
10 right there, but it shows you a lot. It shows you -- let's  
11 see.

12                  You gave me a blue one, too.

13                  Let's see. It shows you here we have Public Class --  
14 oh, no. Package java.lang Public Class Math, Public static int  
15 max. So here we have the fully qualified name of this method.  
16 It's java.lang.math.max. That's the name and the rest of the  
17 API is that it returns an integer and it takes two integers as  
18 arguments.

19                  You'll notice that I haven't circled the names here  
20 (indicating). They are called the formal parameter names.  
21 They are actually used just inside here, so we can actually do  
22 the calculation. And that's it. That's an API and its  
23 implementation.

24                  So the stuff that I've sort of circled and underlined  
25 in blue, that's the API. And this stuff that I'm now putting

1 in the big square box, that's the implementation.

2           **THE COURT:** When you say it returns, in your example,  
3 arg2, is that given a name like X or Y or --

4           **THE WITNESS:** That's are the names. In algebra these  
5 are the variable names, you know, like A squared plus B squared  
6 equal C squared.

7           So I'm just basically making some variable names.

8 I'm saying there is a variable called arg1 of type int and arg2  
9 of type int.

10          Why do I have to name the arguments? There is really  
11 only one reason I have to name them, which is so that I can  
12 refer to them in the implementation. The names are barely part  
13 of the API.

14          **THE COURT:** How do you know when you go back which  
15 one of them it's returning?

16          **THE WITNESS:** I'm sorry. I don't understand the  
17 question.

18          **THE COURT:** In your example of one and 13.

19          **THE WITNESS:** Yeah. Well, notice that I have given  
20 the two arguments names. I call them arg1 and arg2. And when  
21 I call them that, an invocation of the method -- I'm going to  
22 come back to this page in a second.

23          If I called them up there, I might say something like  
24 int x equals math.max (1, 13).

25          That's called a method invocation and it makes the

1 computer actually execute this code. So what does the computer  
2 do? First it assigns 1 to arg1 and 13 to arg2.

3           **THE COURT:** I see. So go back to the other page.

4           **THE WITNESS:** Fine.

5           **THE COURT:** In the next step going forward "x" will  
6 then be the greater of those two numbers?

7           **THE WITNESS:** It will after the method has done its  
8 job.

9           **THE COURT:** I see. All right. You answered my  
10 point.

11           **THE WITNESS:** And this actually is a local  
12 demonstration. It shows you how APIs work.

13           We don't have to worry about the details, what's  
14 going on underneath this function. We have this little  
15 language. If I want to get the maximum of the two numbers, I  
16 just asked for it through max. It does the work and I don't  
17 have to worry about how it does the work.

18           Should I return to the witness stand now?

19           **THE COURT:** Yes. Thank you.

20 **BY MR. BABER:**

21 **Q.** Stay right there a moment, Dr. Bloch.

22           On your chart where you put the package name at the  
23 top, the method declaration below it and the implementing code  
24 in the blue box that you have in front of you, would a source  
25 code file such as that typically also include anything else in

1 the way of comments, prose?

2 **A.** Yes, it would. It wouldn't fit on this sheet, especially  
3 not with my sloppy writing. But immediately above the method  
4 you would have in a special kind of comment that looks like  
5 this. Slash star star and then, you know, a lot of words, star  
6 slash.

7 That comment is called a java.comment and it is the  
8 API documentation for that method. It is automatically  
9 processed by a tool called JavaDoc and it makes the API  
10 documentation that if you haven't seen yet in this trial, I'm  
11 sure you will.

12 **THE COURT:** May I interrupt?

13 Using a different color.

14 **THE WITNESS:** I have red and blue. Do you have one  
15 more?

16 **THE COURT:** Dawn, what happened to all of our magic  
17 markers?

18 **THE CLERK:** Yours are in chambers.

19 **THE COURT:** A highlighter or magic marker.

20 (Whereupon, a marker was tendered  
21 to the witness.)

22 **THE COURT:** All I want you to do is identify which  
23 one is the name, which one is the declaration and which one is  
24 whatever else, et cetera.

25 **THE WITNESS:** All right. I'm sorry. Things are a

1 little bit spread out, but I'm going to circle in green the  
2 various components of the name, okay? The name -- you know,  
3 just like your address. Fully qualified name consists of  
4 various parts -- or name; first name, last name, middle name,  
5 right?

6 So the first name of this method is max. Middle name  
7 is the class that it's in, which is math. And the last name,  
8 the clan name, family name is java.lang. You put them all  
9 together and you get java.lang.math.max. And that's the fully  
10 qualified name of the method.

11 **THE COURT:** Would you write "name" there in that  
12 corner?

13 **THE WITNESS:** How about this? Three arrows going to  
14 those three boxes.

15 **THE COURT:** And write the word "name." That's what I  
16 want you to do.

17 (Witness complied.)

18 **THE WITNESS:** And it says "name."

19 **THE COURT:** All right. Now where is the declaration?

20 **THE WITNESS:** All right. The declaration -- now to  
21 be perfectly honest with you, the java.lang spec uses these  
22 terms in a slightly strange way.

23 The language spec says this is all the declaration,  
24 but this (indicating) they call the signature. But most  
25 computer people just call this the declaration. So I'm going

1 to use the common parlance --

2           **THE COURT:** You did use "signature" with Mr. Jacobs.

3 I was going to ask you what that means. So show us --

4           **THE WITNESS:** Signature and declaration are the same  
5 except, once again, strange terminology in the java.lang  
6 specification. The signature does not include the return type  
7 technically, and I don't know why. It's just the way it is.

8           **THE COURT:** You're confusing me now.

9           **THE WITNESS:** I'm sorry. I apologize. Let's forget  
10 about that technicality.

11           The signature or declaration is this (indicating).  
12 Public -- sorry. I want to do the method declaration. So I'm  
13 going to circle just the method declaration.

14           Gosh. Do we have any more colors?

15           **THE CLERK:** Black.

16           **THE WITNESS:** Black is good. Thank you.

17           (Whereupon, a marker was tendered

18           to the witness.)

19           **A.** All right. So in black here what I'm going to do is  
20 circle the method entire declaration. So it says, Public  
21 static int max int arg1 int arg2. I kind of wish this had fit  
22 on one line, but this tells you how to call the method. It  
23 tells you the short name that you use. The first name, like,  
24 Josh is max. And it tell you what argument types to pass in  
25 and what argument type you will get out.

1                   **THE COURT:** Write the word "declaration" somewhat in  
2 a place that we can remember.

3                   **THE WITNESS:** I will put it in the box. I'm sorry my  
4 writing is so sloppy.

5                   And then this is implementation. So I should label  
6 that as -- oh, orange. That's great.

7                   **THE COURT:** They gave you a highlighter. A  
8 highlighter is not a magic marker.

9                   (Laughter.)

10                  **THE COURT:** I can't read that. I can't read that.  
11 Thank you. That's helpful. Please continue on.

12                  How much longer do you have?

13                  **MR. BABER:** I would say -- let me look quick, your  
14 Honor. Thirty minutes; 20, 30 minutes?

15                  **THE COURT:** Let's try to go to -- it's too soon for a  
16 break, but within 10 minutes we need to reach arresting point.  
17 And take our next break, but go ahead. I don't want to do it  
18 quite yet.

19                  **BY MR. BABER:**

20                  **Q.** Dr. Bloch, during your last discussion of the method  
21 declaration or method signatures, I just want to make sure two  
22 terms are clear.

23                  You referred to an argument. What is an argument in  
24 the context of an API?

25                  **A.** An argument is the same as a parameter. So they are the

1 values that you pass in to the method. In this case they are  
2 arg1 and arg2.

3 **Q.** Even a simpler basis, Dr. Bloch. Is an argument what the  
4 programmer provides to the API? It's the input to the API?

5 **A.** Yes, yes.

6 **Q.** Okay. You also used the word return. What is the return  
7 in the context of an API?

8 **A.** So it's the opposite. It's what the API provides to the  
9 caller.

10 **Q.** So an argument goes in and the return comes back?

11 **A.** Yeah. So by our contract analogy, the argument is the  
12 money and the return is what you're getting for the money. How  
13 about that?

14 **Q.** So, Dr. Bloch, we were just starting to talk about in your  
15 days at Sun where, in addition to designing APIs, you also were  
16 tasked with reimplementing or improving existing APIs; do you  
17 recall that?

18 **A.** I do.

19 **Q.** Now, in that context when you are working with an existing  
20 API that has already been created and made public, are there  
21 any constraints on the creativity that you can exercise with  
22 respect to reimplementing that API?

23 **A.** Constraints on the creativity? Yeah. It's highly  
24 constrained; that is, if you're reimplementing it, there is no  
25 creativity involved any more.

1 Q. Okay. When you say there is no creativity involved, are  
2 there any parts of the API itself, of the method declaration,  
3 the package that it's in, any of those elements that you could  
4 change when you are reimplementing an existing API?

5 A. There is only one.

6 Q. What's that?

7 A. You can change the argument names.

8 Q. So in your example from your chart of the max method, you  
9 don't have to call variable one, in effect, arg1, you could  
10 call it "A" or "X."

11 A. You could.

12 Q. You can call it whatever you want.

13 A. But if you change the name in the API, you also have to  
14 use the new name in the implementation.

15 Q. But you can't change the package or the class?

16 A. No.

17 Q. You can't change the method name?

18 A. No.

19 Q. You can't change the return type, the output as being an  
20 integer?

21 A. No. Except in esoteric situations.

22 Q. And you also can't change the fact that there's two  
23 arguments? Two pieces of data have to go in, correct?

24 A. Correct.

25 Q. And that they both have to be integers?

1     **A.**    Correct.

2     **Q.**    If you are reimplementing an existing API, do you have  
3 creativity with respect to the implementing code, what's down  
4 here in your orange box?

5     **A.**    Yes, you have complete creativity.

6     **Q.**    Does the API limit you in any way as to how you can write  
7 that code to perform the function that's required?

8     **A.**    No. There are virtually no limits on it.

9     **Q.**    Does the API tell you how to write that code?

10    **A.**    No, it doesn't. And even in something as simple as this,  
11 math.max, there is more than one way to do it.

12              **THE COURT:** May I ask a question on that?

13              Let's say you had exactly that scenario. You had two  
14 versions, I guess is the word, that had the implementations  
15 written in a different approach.

16              **THE WITNESS:** Uh-huh.

17              **THE COURT:** When it got compiled, though, when those  
18 two get compiled in the ones and zeros, are they still going to  
19 wind up being identical or will they have different object  
20 code?

21              **THE WITNESS:** Typically they will have very, very  
22 different object code. If it is simple enough, if it's a tiny  
23 little function, you might be able to have two different  
24 implementations that end up emitting the same object code. It  
25 could happen.

1           And I could even give you an example of another way  
2 to implement max that is likely to generate the same object  
3 code. But that's definitely the exception rather -- that's the  
4 rare case.

5           **THE COURT:** Okay. Thank you.

6           **BY MR. BABER:**

7           **Q.** Okay. Now, Dr. Bloch, when you were at Sun prior to the  
8 time you joined Google in 2004, did you have any experience  
9 with respect to independent implementations of existing Java  
10 APIs.

11           By "independent" I mean implementations that were  
12 being done by someone other than Sun?

13           **A.** Before I joined Sun --

14           **Q.** I'm sorry. Before you joined Google, while you were at  
15 Sun.

16           **A.** Oh, yes. At Sun several times.

17           **Q.** Okay. And what third-party implementations of Java APIs  
18 did you have involvement with?

19           **A.** I helped a junior engineer by the name of Mike McCloskey.  
20 He actually goes by the name of Mad Bot. I helped him add  
21 regular expressions to the Java programming language.

22           And for the regular expression APIs he wanted to use  
23 the most common API at the time, which was perl's regular  
24 expression API.

25           **Q.** Okay. I don't want to interrupt you, but I think you may

1 have misunderstood my question, okay. So let me back up and do  
2 it in a different order.

3 **A.** Okay.

4 **Q.** First, while you were working at Sun, did you reimplement  
5 some existing APIs for Sun internally at Sun?

6 **A.** Yeah. I did it a couple times. That was one of them. If  
7 you want to cover them in chronological order, we can do that.

8 **Q.** Well, just give the jury an example of one class that you  
9 worked on while you were at Sun where the API was already  
10 existing and you were simply reimplementing it or writing new  
11 code for it?

12 **A.** All right. So there was something called big integer in a  
13 package called java.math. And that lets you work with really  
14 big numbers that can have 100 digits.

15 I actually wrote that API in the Version 1.1. of the  
16 Java Platform, but back then Java was really slow. So instead  
17 of having the implementation be in Java -- the implementation  
18 needs to what are called native methods, called C Code, which  
19 is another program language, the C programming language, which  
20 at the time was much faster than Java. So the implementation  
21 was a veneer. It was just Java that called C.

22 **Q.** Is that just one example of any number of occasions in  
23 which you reimplemented --

24 **A.** Actually, I haven't told you about the reimplementation  
25 yet. That was the original implementation.

1           So by Java 1.3 the language -- the implementation of  
2 our program had gotten so much faster that we could write the  
3 whole thing in Java and have it run as fast or faster than it  
4 ever ran before.

5           I helped Mike McCloskey reimplement the entire  
6 `Java.math.integer` in the Java, the version that you're running  
7 today. If you run Java, that is the one Mike McCloskey wrote  
8 and it's a complete reimplementation of our API.

9 **Q.** On any of the occasions while you were at Sun when you  
10 worked on reimplementing an existing API, did you ever change  
11 any of the elements of the method declaration for an existing  
12 method?

13 **A.** No. We couldn't.

14 **Q.** Why couldn't you?

15 **A.** Because it wouldn't work any more, because programs that  
16 had been written to use that API would no longer work. You  
17 would compile them and there would be a mismatch. You would  
18 call a method name, the method name better be the same. If you  
19 change the name, the program won't work any more. It would be  
20 an incompatible change.

21 **Q.** And in addition to working on reimplementations of APIs  
22 internally at Sun, were there other occasions when you also  
23 were involved in reimplementations of APIs that were occurring  
24 outside of Sun?

25 **A.** Right. I misunderstood your previous question. Right.

1           So this -- this is the case of regex, which is also  
2 regular expressions. This was also done by Michael McCloskey.

3           In this case when we were adding regular expressions  
4 to Java, we had to choose an API to describe these regular  
5 expressions, but it turns out that people were already using  
6 another API to describe their regular expressions and it came  
7 from a language called Perl, P-E-R-L. And that scripting  
8 language called Perl had become very popular.

9           So many, many people already had millions of regular  
10 expressions written in this exact syntax, and we wanted to make  
11 sure that our implementation was consistent with, was  
12 compatible with this existing syntax. So we reimplemented the  
13 Perl regular expression API in Java and that is the Java  
14 regular expression API which is java.util.regex. That's the  
15 name of the package.

16 **Q.** Just so we're clear and the jury understands, Dr. Bloch,  
17 java.util.regex, R-E-G-E-X, is an example of classes in the  
18 Java API that Sun got from someone else and reimplemented and  
19 put them in the Java package, is that right?

20 **A.** Yes. We got the API from somewhere else, but we did our  
21 own independent implementation. We did not copy the code.

22 **Q.** And I believe you said that that API came from the Perl  
23 scripting language?

24 **A.** It did. And, in fact, we were able to make use of 30,000  
25 existing tests that checked to see if a Perl implementation was

1 obeying its specification. We ran those same tests on the Java  
2 implementation and it passed. So we had great confidence that  
3 our Java implementation was good.

4           **THE COURT:** Is this a good breaking point or do you  
5 want to ask a couple more questions?

6           **MR. BABER:** I've got about three more questions, your  
7 Honor, and then it would be a good break.

8           **THE COURT:** Go ahead.

9           **BY MR. BABER:**

10          **Q.** In addition to implementing existing Java APIs and then  
11 implementing outside APIs and bringing them into the Java  
12 libraries, were there also occasions when you were working at  
13 Sun when you worked with someone else who was creating a set of  
14 Java class libraries and implementing the existing APIs?

15          **A.** Yeah. So there was this project called Classpath. It was  
16 an Open Source project. And they were doing an early Open  
17 Source implementation of the JDK.

18           And occasionally, after I had written the first  
19 version of the collections library that I referred to before,  
20 these folks from the outside would send an email saying, "Hey  
21 we're trying to reimplement your libraries, but we don't  
22 understand your specs."

23           In this corner case (indicating), you know, let's  
24 say, "I'm asked to find the maximum element of this list, but  
25 the list has no elements in it. What should I do?" And I

1 would say, "Oh, darn. I forgot to specify that." "It should  
2 throw no such elementec session." "Thanks for helping me fix  
3 the spec." And then I would put that fix, that change into the  
4 specification.

5 **Q.** Okay. And do you recall, Dr. Bloch, did you speak with  
6 the folks from Classpath on more than one occasion as they were  
7 trying to reimplement the collections library?

8 **A.** I don't know if I spoke with them, but I certainly emailed  
9 back and forth, yes, several times.

10 **Q.** The fact that you were doing that, was that known to your  
11 supervisors at Sun?

12 **A.** Yes. David Bowen was my manager and he was aware.

13 **Q.** At the time you did that, you were at Sun and Classpath  
14 was working on an independent implementation of the libraries,  
15 correct?

16                   **MR. JACOBS:** Objection. Leading, your Honor.

17                   **THE COURT:** Wait. Don't -- rephrase the question.  
18 Sustained.

19 **BY MR. BABER:**

20 **Q.** All right. At the time you had these communications with  
21 Classpath, did you believe that the Classpath project was of --

22                   **THE COURT:** Sustained.

23                   **MR. BABER:** I'm sorry.

24                   **THE COURT:** That's leading. The way to solve leading  
25 is always use "What, if any." Just stick that into the same

1 question, "What, if any," or "To what extent, if at all."  
2 Those two are the end-all or cure-all of all leading problems.

3 **MR. BABER:** Works for me, your Honor.

4 (Laughter.)

5 **BY MR. BABER:**

6 **Q.** Dr. Bloch, what, if any, beneficial effects did you think  
7 that the Classpath project had on the Java APIs?

8 **A.** It helped me ensure that the APIs were of sufficient  
9 quality that they would, first of all, admit independent  
10 implementations and, second of all, allow programmers to use  
11 them and to know exactly what they would do.

12 If an API specification is good enough that someone  
13 else can write a compatible independent implementation, it's  
14 also good enough that a programmer knows exactly what that API  
15 is going to do.

16 **MR. BABER:** This is a good breaking point, your  
17 Honor.

18 **THE COURT:** Thank you. 15 minutes. Remember the  
19 admonition.

20 **THE CLERK:** All rise.

21 (Jury exits courtroom at 11:17 a.m.)

22 **THE COURT:** Please be seated.

23 All right. Any issues for the Court?

24 **MR. JACOBS:** No, your Honor.

25 **MR. BABER:** No, your Honor.

1                   **THE COURT:** We'll take 15 minutes ourselves.

2                   (Whereupon there was a recess in the proceedings  
3                   from 11:18 a.m. until 11:30 a.m.)

4                   **THE COURT:** All right. Please bring the jury back.

5                   **MR. BABER:** Ready, your Honor.

6                   **THE COURT:** The witness is here. Let's bring back  
7 the jury.

8                   (Jury enters the courtroom at 11:31 a.m.)

9                   **THE COURT:** Okay, Welcome back. Please be seated.

10                  I need to ask the team over there to be seated and no  
11 distractions, please.

12                  Counsel, the floor is yours.

13                  **MR. BABER:** Thank you, your Honor.

14                  **BY MR. BABER:**

15                  **Q.** Dr. Bloch, I would like to talk a little bit about your  
16 time at Google. You left Sun and joined Google in what year?

17                  **A.** 2004.

18                  **Q.** And during your first several years at Google, just  
19 generally, what did you do?

20                  **A.** I did Java infrastructure within Google. So I took the  
21 sort of pieces of Google's internal infrastructure. And when I  
22 arrived, they were accessible only from within the C++  
23 programming language. You had to sort of jump through hoops to  
24 use them from Java programs.

25                  My group made it easy to use these infrastructure

1 pieces from within Java. We wrote APIs for them and then we  
2 wrote implementations for those APIs. Sometimes the  
3 implementations just talked to the existing C++ implementations  
4 and sometimes they were brand new from-the-ground-up  
5 implementations.

6 **Q.** And at some point in time you joined the Android team, is  
7 that right?

8 **A.** Yes.

9 **Q.** And when was that again?

10 **A.** I believe that it was at the end of 2008 or early in 2009.  
11 It was between December 2008, January, 2009.

12 **Q.** Dr. Bloch, what we have here is a timeline of agreed-upon  
13 dates. If you joined the Android team in late 2008, had  
14 Android already been released?

15 **A.** Yes. Yes, it had.

16 **Q.** The timeline shows November 2007 Google releases the  
17 Android software development kit. That had happened a year  
18 earlier. Is that your recollection as well?

19 **A.** That is my recollection.

20 **Q.** And the Android phone started coming out on the market in  
21 late 2008, both the HTC Dream and -- I guess, the HTC dream  
22 that year?

23 **A.** Yes.

24 **Q.** So when you joined the Android team in late 2008 or early  
25 2009, what was the status of the Java language libraries that

1 were part of the Android runtime?

2 **A.** They worked. They were largely complete, but they weren't  
3 necessarily as fast as they could be and perhaps they had more  
4 bugs than they should have.

5 **Q.** And did you have any role prior to joining the Android  
6 team in any decisions about what packages or methods or classes  
7 to include in Android?

8 **A.** None whatsoever. I simply wasn't involved with the  
9 Android team.

10 **Q.** Now, how long did you remain on the Android team?

11 **A.** Just about a year.

12 **Q.** And what did you do as a member of the Android team during  
13 that time?

14 **A.** I wrote implementations for core Java libraries. I fixed  
15 bugs in existing ones and occasionally rewrote them entirely.

16 **Q.** Your talking about the libraries that were in Android?

17 **A.** Yes.

18 **Q.** The Java libraries?

19 **A.** Yes, the libraries that were in Android.

20 **Q.** And during that year that you worked on the Android team  
21 in connection with any work that you were doing on the Android  
22 core libraries, did you consult any Sun source code or other  
23 Sun materials?

24 **A.** Not that I'm aware of.

25 **Q.** And in your work with these APIs that were already in

1 Android where you were working on the code and trying to  
2 improve it and fix the bugs, as to any of them, did you ever  
3 make any changes to any of the method declarations or the other  
4 elements of the APIs that you identified in the chart that you  
5 drew this morning?

6 **A.** None whatsoever. I couldn't. It wouldn't have been  
7 possible.

8 **Q.** Okay. Now, Dr. Bloch, Mr. Jacobs asked you about a file  
9 by the name of Timsort; do you recall that?

10 **A.** I do.

11 **Q.** And what is Timsort? What does Timsort do?

12 **A.** Timsort simply takes a list of things, usually numbers and  
13 strings, and puts them in order; either alphabetical order or  
14 numerical order.

15 **Q.** Let me just cover quickly and I think we can put it aside  
16 for a second, but is there also a second file named Comparable  
17 Timsort?

18 **A.** Yes. The idea behind Comparable Timsort is it let's you  
19 specify the order. Suppose for some strange reason that you  
20 want to sort strings not in alphabetical order, but by how long  
21 they are with the shortest words, like "a" and "I" at the  
22 beginning and the longer ones like  
23 "Antidisestablishmentarianism" at the end.

24                   And I apologize to the court reporter for doing  
25 that.

1 (Laughter.)

2 Q. And with respect to the rangeCheck function that  
3 appears -- does the rangeCheck function appear in both Timsort  
4 and Comparable Timsort?

5 A. Yes, it does.

6 Q. At least when you first wrote them?

7 A. It did.

8 Q. And with respect to the rangeCheck function, are there any  
9 differences between Timsort and Comparable Timsort?

10 A. No, not with respect to the rangeCheck function.

11 Q. Now, about how many lines of code are in Timsort?

12 A. About 900.

13 Q. And prior to your creating the Timsort file, was there an  
14 earlier file that did the same thing, basically the same  
15 function that was already in the Java APIs?

16 A. It didn't deserve its own file, but there was an  
17 implementation of that API and it was in a file called a  
18 arrays.java.

19 Q. Okay. And how did Timsort, the file you created, compare  
20 to arrays.java in terms of performance?

21 A. It was much, much faster, especially for arrays that were  
22 somewhat ordered to start with.

23 Q. And when you say "much faster," can you quantify that at  
24 all?

25 A. It depends on the inputs. Seeing a speed up 20 times

1 faster was not uncommon. I've seen things even more than that  
2 and, you know, twice as fast is a good sort of overall estimate  
3 if you average everything.

4 **Q.** Okay. Now, within that Timsort file, among the 900 lines  
5 of code, is there a method named rangeCheck?

6 **A.** There is. It's at the very end.

7 **Q.** What kind of method is rangeCheck?

8 **A.** It's a very short simple method. In fact, it's very  
9 similar to that method that I drew on that chart up there. If  
10 the jury compares rangeCheck with that, they will see that they  
11 are very similar.

12 **Q.** Is rangeCheck a Private method or a Public method?

13 **A.** It's a Private method.

14 **Q.** And what does that mean, that rangeCheck is a Private  
15 method?

16 **A.** It means that it's not part of the API. It has a  
17 declaration, but that declaration is used only from inside the  
18 Timsort class.

19                   Outside of the Timsort class it simply doesn't exist.  
20 You cannot access that rangeCheck function as a user of Java.

21 **Q.** If a programmer is writing a program and they want to  
22 check some things on a list, they can't type in -- like they  
23 could type in "max" to get the higher of two numbers, they  
24 can't type in "rangeCheck" and have some checking be performed?

25 **A.** Nope. They've got to write their own.

1     **Q.** Now, you mentioned that Timsort performed much faster than  
2 the old arrays.java file. Was any of the faster performance of  
3 Timsort due to the rangeCheck method?

4     **A.** No.

5     **Q.** Okay. And what, what exactly does the code in the  
6 rangeCheck method do?

7     **A.** So when you're sorting a list, you can either sort the  
8 whole list or you can sort just a part of the list. And if  
9 you're sorting just a part of the list, you have to say sort  
10 from here, a third position, to here, the eighth position. And  
11 before you even start sorting, it's imperative that you check  
12 that three and eight are actually within the list and in the  
13 right order.

14                 If the list only contains 10 things and you ask to  
15 sort the hundredth through thousandth element, you can't sort  
16 it. The best you can do is tell the programmer, sorry, you  
17 goofed. That's called throwing an exception. So that's  
18 exactly what rangeCheck does. It looks at that first index,  
19 the three. The last one, the eight. And first it says, "Hey,  
20 is three less than zero?" Because if it's negative, it's  
21 invalid. And what about eight? Is eight greater than the size  
22 of the list? If it's greater than the size of the list, you're  
23 trying to access an element that's outside of the list, so it's  
24 invalid. In either of those cases you throw an exception.

25                 And finally you check if they are in the wrong order

1 because computers are very, very literal. So if I ask to sort  
2 the eighth through third instead of the third through eighth,  
3 it's violating my half of the contract and the computer doesn't  
4 know what to do, and the best it can do is to throw an error.

5 So what rangeCheck does, then, is it simply makes  
6 these three checks, makes sure that the numbers, the indexes  
7 are valid. If they are valid, great. It does nothing. If  
8 they are invalid, it throws an exception. It tells the  
9 programmer, "I can't sort your list."

10 **Q.** Other than checking those three things, does rangeCheck do  
11 anything else?

12 **A.** Nothing.

13 **Q.** And is rangeCheck a simple, complex? How would you  
14 describe the function, the rangeCheck function?

15 **A.** Very, very simple. Any competent high school programmer  
16 could write it.

17 **Q.** Now, are you familiar, Dr. Bloch, with any earlier files  
18 that included a function named rangeCheck that performed these  
19 same three checks?

20 **A.** Yes. I have an exhibit in front of me that shows me that  
21 the arrays.java file contained a very similar function.

22 **Q.** And do you know when the rangeCheck function in the  
23 arrays.java file was created?

24 **A.** I think I wrote it sometime around 1997.

25 **Q.** So you wrote the rangeCheck function that appears in

1 arrays.java?

2 **A.** I believe I did.

3 **Q.** Okay. Now, when did you begin working on Timsort?

4 **A.** I believe that it was around July, 2007.

5 **Q.** And did you start from scratch with Timsort or did  
6 Timsort --

7 **A.** No, no. Timsort -- sorry for the interruption.

8 Timsort is what is known as a port. There's a reason  
9 it's called Timsort. There actually is a Tim. His name is Tim  
10 Peters and he wrote this wonderful fast sorting routine for a  
11 language called Python. And I didn't know about it, but I'm  
12 friends with Guido Van Rossum, who is the father of Python, the  
13 same that James Gosling is the father of Java.

14 And Guido said to me one day, "We have great sort  
15 called Timsort. You should try it in Java." I said, "Okay."  
16 And I took it and I did what's called porting it to Java, where  
17 you take code that's written in one programming language and  
18 you kind of translate it so that it works in another  
19 programming language.

20 It was just kind of an experiment for my own  
21 purposes. I think I may even have started doing this at home  
22 on my computer.

23 **Q.** Okay. You mentioned just a minute ago, Dr. Bloch, that  
24 you got -- you got some code from someone, is that right?

25 **A.** Yes. This code came from a language called Python.

1 Q. Where did -- from whom did you actually get the code that  
2 was sent -- that you obtained?

3 A. I downloaded from the Python repository.

4 Q. And did you have any understanding as to whether or not  
5 that code was Open Source, available for use?

6 A. Yes. Guido told me that it was -- it was Open Source and  
7 it was licensed under a permissive Open Source language --  
8 sorry. Open Source license, which in Guido's words said -- it  
9 meant you can do anything you want with it. Just take it.  
10 Translate it to Java. If you like it, use it.

11 Q. And this was in, I think you said, the middle of 2007?

12 A. Yes.

13 Q. And you were not a part of the Android team at that time,  
14 right?

15 A. Oh, no, because it was over a year.

16 Q. You were still working on the infrastructure projects at  
17 Google?

18 A. Yes, I was.

19 Q. And so when you started working on Timsort in the middle  
20 of 2007, what did you plan to do with it if it worked and you  
21 liked it?

22 A. Oh, if it was fast, I wanted to put it back into the Open  
23 JDK so that everybody could use it. Without changing their  
24 programs at all, they would just run faster.

25 Q. And when you say "put it back," what do you mean exactly

1 because there was no Timsort in Open JDK.

2 **A.** No, no, no. "Put back" is a term of art. What I mean is  
3 I wanted to contribute it to the Open JDK.

4 **Q.** And what is the Open JDK?

5 **A.** Open JDK is Sun's, now Oracle's, open implementation of  
6 the Java SE platform.

7 **Q.** And does anyone control Open JDK in the sense of they have  
8 to approve things that get contributed to it in order to  
9 include them?

10 **A.** Yes.

11 **Q.** And who controlled Open JDK in 2007, 2008?

12 **A.** I think it's fair to say that Sun controlled it.

13 **Q.** Okay. And now prior to the time you started working on  
14 the Timsort file in 2007, had you contributed anything to Open  
15 JDK before that?

16 **A.** Oh, yes.

17 **Q.** And as a general matter, how would you contribute things  
18 to Open JDK? I mean, physically how did that work?

19 **A.** It actually differed over time. Small contributions I  
20 usually gave to my friend Professor Doug Lee at the State  
21 University of New York at Oswego. He has his own source  
22 repository and I would put stuff into Doug's.

23 **Q.** Let me stop right there. Can you explain to the jury,  
24 please, what is a source repository?

25 **A.** Sorry. When you have a big project like a Java

1 implementation, you have all of this programming, all of the  
2 source code that implements Java, all the code that makes Java.  
3 You store that code somewhere, and that's called a repository.

4 And Doug Lee, who is this professor, has a repository  
5 because he runs what's called JSR 166x. He has this sort of  
6 ongoing 10 year-long JSR where we just do contributions to  
7 things like Java.util.concurrent. That's its sort of formal  
8 purpose, but we slip things into Java.util as well.

9 For instance, in the previous release, which is  
10 called Java 6, I contributed a bunch of new collection  
11 implementations to Open JDK. They had names like ArrayDec.  
12 And I don't know. I don't remember the other ones. So that  
13 was one way of contributing.

14 And larger things, like Timsort, were contributed  
15 somewhat more formally through the actual Open JDK process.

16 Q. You actually submitted to Sun?

17 A. Yes. I submitted it directly to Sun.

18 Q. Okay. And, Dr. Bloch, if you were an employee of Google  
19 at the time, why would you contribute to Sun's Open JDK  
20 software?

21 A. Java is a large part of my life, a large part of my  
22 career. You know, I kind of made my reputation in Java and I  
23 feel like Java gave me a lot and I owe Java a lot back. So I  
24 contribute to the Java ecosystem whenever I can, and I have  
25 been doing that for years.

1   **Q.** Now, Dr. Bloch, when you were working on the Timsort file,  
2 were there any technical reasons why it made sense to use the  
3 same rangeCheck function in Timsort as the one that already  
4 existed in arrays.java?

5   **A.** Yeah.

6   **Q.** What were the reasons?

7   **A.** Timsort got directly into arrays.java and then you could  
8 simply throw away the extra rangeCheck. I wanted Timsort to  
9 simply call the existing rangeCheck function that was in  
10 arrays.java. And that's good software engineering for a couple  
11 of reasons.

12   **Q.** Now, at the time that you were working on the Timsort  
13 file, were you working with it as part of all the rest of  
14 arrays.java or were you just working on the Timsort part?

15   **A.** No. I was working on it alone. Actually, working on the  
16 core libraries is a complicated process, but this is pretty  
17 much a free-standing sort, so I could just work on it on my  
18 laptop. You know, it was not in any package at all.

19   **Q.** And at the time you began working on Timsort, did you have  
20 an expectation as to what would happen ultimately with the  
21 rangeCheck method that was one of the Public methods in the  
22 Timsort file?

23   **A.** Public methods?

24   **Q.** I'm sorry. Private methods.

25   **A.** Private methods. All right. So, yeah. No, I expected it

1 would simply go away. I didn't realize quite how large Timsort  
2 was going to get, so I thought it would just get dumped into  
3 arrays.java. I thought the two files would be merged and  
4 Timsort would be a Private class within the Timsort class and  
5 actually have little classes that never even escape from the  
6 file. So that was the plan.

7                   But eventually it became clear that Timsort was just  
8 too big to put into arrays, which was already an extremely  
9 lengthy file.

10 **Q.** And I believe you told us you joined the Android team  
11 either at the end of 2008 or the start of 2009.

12 **A.** That's correct.

13 **Q.** Now, had you finished the Timsort file when you joined the  
14 Android team?

15 **A.** Not really. I had gotten to the point where it ran really  
16 fast. It was great. And I knew that I was going to be  
17 contributing it back. But, you know, I hadn't finished it.

18 **Q.** Okay. Now, at the time you joined the Android team, was  
19 the rangeCheck function already in the Timsort file?

20 **A.** Oh, yes.

21 **Q.** Okay. Do you recall specifically when you added the --  
22 what point in the process you added the rangeCheck function to  
23 the Timsort file?

24 **A.** I would assume as soon as I started writing it, you know,  
25 basically when I was going to write Timsort because I knew that

1 it was headed for inclusion in arrays.java. You know, it was  
2 part of the scaffolding, so I think it was probably the first  
3 thing I put in there.

4 **Q.** Approximately when did you finish Timsort?

5 **A.** Gosh. Probably in early 2009, I don't remember the month.

6 **Q.** And when you finished with Timsort, what did you do with  
7 it?

8 **A.** I contributed it back to the broader Java ecosystem, to  
9 the Open JDK and to Android.

10 **Q.** And was Timsort accepted for inclusion in the Open JDK?

11 **A.** Yes, it was.

12 **Q.** And was it, in fact, ultimately added to the Open JDK?

13 **A.** Yeah. If you download Java 7 today, you'll get it.

14 **Q.** What is Java 7?

15 **A.** Java 7 is the most recent release of the Java SE Platform.

16 **Q.** And was Java 7 announced in or around late 2009?

17 **A.** You know, it was preannounced long before that, but, yes.

18 **Q.** Do you recall any discussion of Timsort being added to  
19 Java 7 that was in around November of 2009?

20 **A.** It was like the milestone 5 build that it got into.

21 **Q.** So the milestone 5, build of Java 5 included Timsort?

22 **A.** It did.

23 **Q.** Okay. And at the time the milestone 5 build was  
24 announced, did anyone from Sun or Oracle make any statements  
25 specifically about Timsort?

1     **A.** Yes, they did.

2     **Q.** Who specifically made a statement about Timsort?

3     **A.** I believe that Mark Reinhold praised it for its speed.

4     **Q.** Is that Mark Reinhold, the chief Java architect at Sun,  
5 now Oracle?

6     **A.** That's him.

7     **Q.** Do you recall exactly what Dr. Reinhold said about  
8 Timsort?

9     **A.** I don't recall his exact words, but I could look them up.  
10 They are on his blog.

11    **Q.** Now, is Timsort the only thing you contributed to Java 7?

12    **A.** No, not by a long shot.

13    **Q.** Okay. Can you identify something else you contributed to  
14 Java 7?

15    **A.** Yeah. My biggest contribution to Java 7 was something  
16 called arm blocks, or Automatic Resource Management Blocks.  
17 And that's actually a language feature that I contributed.

18               And, also, I contributed the APIs that are required  
19 to support that language feature.

20    **Q.** All right. And in very simple terms, Dr. Bloch, what does  
21 ARM Block do? Just general purpose?

22    **A.** All right. So when you open a file, you have to remember  
23 to close it. If you forget to close it, then that's very bad.  
24 Your program does -- it leaks resources and it can end up just  
25 keeling over. It can kill a server.

1               This automatically closes the files for you in the  
2 simplest of terms. That's what it does.

3 **Q.** Now, you mentioned a minute ago that ARM Block is not just  
4 new APIs and libraries, but it's also a new language feature,  
5 is that right?

6 **A.** Yes, it is.

7 **Q.** Why do you call it a language feature?

8 **A.** Because it actually adds to the syntax of the language.  
9 It's part of the exception handling facilities in Java.

10 **Q.** Did the inclusion of the ARM Block technology that you  
11 created in Java cause any changes to the grammar of the Java  
12 programming language?

13 **A.** Yes.

14 **Q.** You contributed to Open JDK, both the language feature and  
15 the new APIs; is that right?

16 **A.** I did.

17 **Q.** Now, is it common or uncommon, Dr. Bloch, for the language  
18 in APIs to change sort of in tandem?

19 **A.** It is very common. They often change in lockstep for a  
20 few reasons.

21               First of all, because if you add a new language  
22 feature and that language feature depends on an API -- for  
23 example, these ARM Blocks depend on a new interface called  
24 Closable -- you have to add that interface to the API at the  
25 same time as you add the new language feature.

1           Also, just because of the release process, they tend  
2 to batch them out. They come at the same time.

3   **Q.** And over time since the First Edition of the Java Language  
4 Specification was released back in 1990's, approximately how  
5 many occasions have there been changes to the language?

6   **A.** If you don't count the initial release, I think there  
7 would be six, if I don't miss my guess. I could count.

8   **Q.** Dr. Bloch, do you know whether or not the rangeCheck  
9 method, the nine lines of code that were in Timsort, are still  
10 in the current release of Android that's publicly available on  
11 the Android website?

12   **A.** I know that they aren't. That release is called Ice Cream  
13 Sandwich. No -- yeah, something like that.

14   **Q.** How do you know?

15   **A.** Because I checked.

16   **Q.** And do you know how long it has been gone from the public  
17 available release of Android?

18   **A.** Over a year, but I don't remember the exact date of the  
19 change.

20   **Q.** Dr. Bloch, just very quickly. We've talked about the  
21 contribution of Timsort to Java 7. We talked about the  
22 contribution of the ARM Block, language features and APIs to  
23 Java 7.

24           Have you made other contributions to Java since you  
25 have been at Google?

1     **A.**    Tons. Even to Java 7. I contributed another sort, which  
2 was called Dual Pivot Quick Sort. It was also really fast, but  
3 I won't talk about it.

4                 Anyway, yeah, as I say, I really feel a personal  
5 responsibility to give back to the Java ecosystem. It made my  
6 career, and I have been acting on that ever since I joined  
7 Google. One of the things first things I did at Google was  
8 wrote documentation for the Java features that I added to the  
9 Java 5 version, which is described in this book (indicating).

10               So some of the prose that made it in here is prose  
11 that I wrote while I was a Google employee.

12     **Q.**    Any way to quantify, Dr. Bloch, the number of  
13 contributions you've made to Open JDK or to the Java Platform  
14 since you have been at Google even?

15     **A.**    There is, but I don't know that number. I haven't looked  
16 at the number of lines of code or whatever.

17     **Q.**    In addition to your contributions, have other employees of  
18 Google also contributed to Java?

19     **A.**    Yes. Google was among the greatest contributors to the  
20 Open JDK.

21                 **MR. BABER:** Your Honor, I pass the witness.

22                 **THE COURT:** Thank you.

23                 Redirect.

24                 **MR. JACOBS:** Can we have 794 up?

25                 (Document displayed)

## **REDIRECT EXAMINATION**

BY MR. JACOBS:

Q. Now, Mr. Block, you gave a lot of technical testimony. I just want to boil it down to its essence.

You're not denying that you accessed Sun copyrighted code for arrays.java when you wrote Timsort.java, correct?

A. I'm not denying it, but I also can't affirm it positively. I simply don't remember doing it.

Q. But you said you're perfectly willing to believe that you did --

A. Given the circumstances under which I wrote the code, yes.  
I'm perfectly willing to believe it.

Q. And your contribution of the same code to JDK, you don't contend that that justifies your copying of arrays.java code into Timsort.java and Android, correct?

**A.** No, I do not. If I did that, it was a mistake and I'm sorry I did it.

Q. Now, you talked about your commitment to Java and moving it forward. You actually sit as a member of the Java Community Process Executive Committee; do you not, sir?

**A.** Yes. I do to this day.

Q. And SE 7 of Java, the most recently approved release, that was a very important milestone in the development of Java, wasn't it?

A. It was pretty important.

1 Q. It was pretty important because it had been delayed a bit,  
2 right?

3 A. Yeah.

4 Q. And isn't it true, sir, that you actually cast the only  
5 "no" vote among the commercial members of the JCP against the  
6 release of SE 7?

7 A. That question doesn't have a "yes" or "no" answer. That  
8 was a very, very complicated vote.

9                  There were three other members who would have cast  
10 "no" votes and they resigned in protest rather than casting  
11 those "no" votes. That left 15 members --

12 Q. I'm sorry, Dr. Bloch. I'm kind of on the clock here, so I  
13 need to ask you if you can -- if you can't answer "yes" or  
14 "no," just say that and I'll do something else.

15                  Did you, sir, cast the only "no" vote against the JCP  
16 Executive Committee's approval of SE 7?

17 A. I can't answer that with a "yes" or "no."

18 Q. Isn't it true, sir, that one of the reasons that there was  
19 a disagreement on the Executive Committee about SE 7 was about  
20 Apache Harmony licensing?

21 A. Yes. That is true.

22 Q. And that is true, sir? You were aware that Apache Harmony  
23 had not gotten a specification and TCK license from Sun, now  
24 Oracle, and that was a dispute that caused some abstentions and  
25 resignations, correct, sir?

1     **A.** I don't know what a specification license is. I would  
2 have said it was a TCK license.

3     **Q.** You would have said?

4     **A.** It was a TCK license. That was -- that is my  
5 recollection.

6     **Q.** That Apache Harmony did not have a TCK license, correct?

7     **A.** Yes. But you mentioned a specification and TCK license --

8     **Q.** I'm with you, sir. I'll go with your vocabulary.

9                      Apache Harmony did not have a TCK license, correct?

10    **A.** That is correct.

11    **Q.** And that meant that there was a dispute between Apache  
12 Harmony and Sun, now Oracle, during the life of that  
13 disagreement about whether Apache Harmony was licensed or not,  
14 correct?

15    **A.** Whether Apache Harmony had a TCK license.

16    **Q.** And whether it could be distributed on commercial mobile  
17 platforms; correct, sir?

18    **A.** That I don't know. I can't attest to that.

19    **Q.** Now, you talked about in my examination on your direct  
20 about the creativity in API design; do you recall that?

21    **A.** I sure do.

22    **Q.** And it's a craft, correct?

23                      Now, in this chart here you boxed in the API in  
24 black. And if I understood your testimony, for this particular  
25 method the API was the declaration: Public static int max int

1 arg1 int arg2; correct?

2 **A.** And, in fact, the rest of the stuff that's pointed to by  
3 arrows. The names are also, I would argue, is part of the API.

4 **Q.** So java.lang.math at the bottom?

5 **A.** Yes.

6 **Q.** Looking at this on a stand-alone basis, do you see  
7 creativity and artistry there, sir?

8 **A.** Not in this one, it's too simple.

9 **Q.** You said it was a lovely example; did you not, sir?

10 **A.** I did, but I forgot the context -- oh now I remember the  
11 context.

12 I did. And the reason I said it was lovely was  
13 because even though it's trivial, even though there is no  
14 creativity in it whatsoever, it displays all of these points on  
15 one flip chart.

16 **Q.** It's not representative then of APIs in terms of its  
17 create active; is that correct, sir?

18 **A.** Some APIs require only this much creativity and some  
19 require more.

20 **Q.** In the collections framework, for example, that's in the  
21 37 packages that are in dispute in this matter, the collections  
22 framework that changed that fellow's life, that required a  
23 great degree of creativity; did it not, sir?

24 **A.** It did. But seeing that it's in the 37 packages -- is,  
25 you know, it's in one, maybe two of those packages.

1 Q. It's in the packages that are in dispute; correct, sir?

2 A. No. It's spread among packages that are and aren't in  
3 dispute.

4 Q. The creativity and artistry --

5 A. For the record, can I say --

6 Q. Let me move on. Again, I have to watch my clock.

7                 The creativity and artistry is not in any single  
8 method declaration; is it, sir?

9 A. It can be. You know --

10 Q. There could be creativity and artistry even in a single  
11 method declaration?

12 A. It can happen.

13 Q. Is it really the creativity and artistry in designing an  
14 API and packaging these together and organizing them in the way  
15 that overall they make sense?

16 A. I couldn't say that, no.

17 Q. Well, where is the creativity and artistry, then, sir,  
18 that you spoke about so eloquently in your presentation?

19 A. It's in obeying simultaneously all of these rules of the  
20 craft that I've described here so as to produce an API, a  
21 language that is easy for the programmer to use.

22                 In the full version of that talk, by the way -- in  
23 fact, if you -- if that dec goes to the end of that talk, there  
24 are some API refactoring examples where I have a very, very  
25 simple API, which is exactly in this book as well -- not in

1 this book, but it's an exhibit. It's called the thread local  
2 API. And to look at it you would think it's that simple.  
3 Simple ain't easy. To get this simple API, I had to think long  
4 and hard.

5 **MR. JACOBS:** Thank you very much, Mr. Bloch.

6 **MR. BABER:** Just A very few, your Honor.

7 **RECROSS EXAMINATION**

8 **BY MR. BABER:**

9 **Q.** Dr. Bloch, Mr. Jacobs, unfortunately, did not have time to  
10 allow you to explain the vote on Java 7 that you cast as a  
11 member of the Executive Committee.

12 Could you please explain why you voted "no" as part  
13 of the Java Community Process?

14 **A.** Well, it's a long story and I'll try to tell it as quickly  
15 as I can.

16 So basically there had been a dispute between Apache  
17 and the JCP -- actually, Apache and Sun for years. Sun had  
18 promised Apache a license, a TCK license, which I'm not even  
19 going to explain what that is. And Apache -- Sun had refused  
20 to deliver on this. And numerous times there had been votes in  
21 the JCP saying, "This is not good," the JCP would have a  
22 resolution saying this violates the rules of the JCP. In fact,  
23 Oracle, proposed one such motion, I think, back in 2007 just  
24 saying that Sun must grant a TCK license to Apache, but Sun  
25 never did that.

1                   And on the eve of this vote, three members of the JCP  
2 actually quit because they didn't feel that they could vote  
3 "yes" on this thing if it meant violating the terms of the JCP.

4 **Q.** And were those three members representatives of companies  
5 or were they individual members?

6 **A.** One was a representative of an organization, the Apache  
7 Software Foundation. And then there were two individuals,  
8 Professor Doug Lee, about whom I spoke before. I would say  
9 he's probably the one man who's contributed the most to the  
10 platform maybe. And the other one was Tim Pyrels (phonetic  
11 spelling).

12 **Q.** Okay. Now, during all of the discussions that you  
13 participated in as a member of the Executive Committee of the  
14 Java Community Process about Apache Harmony, did you ever hear  
15 anyone from Sun or Oracle ever say that they thought that  
16 Apache Harmony had infringed any copyrights of Sun?

17 **A.** I never heard that.

18 **Q.** All right.

19                   **MR. BABER:** Your Honor, I would like to go ahead and  
20 mark as an exhibit the chart that Dr. Bloch made while he was  
21 on the stand as a demonstrative exhibit. It would be  
22 Exhibit 3452.

23                   **THE COURT:** All right, 3452. But it's not in  
24 evidence. It's just marked for the record.

1 (Trial Exhibit 3452 marked  
2 for identification)

3           **THE COURT:** Again, the demonstrative will not be in  
4 the jury room. You might see this again at some point, maybe  
5 at closing argument, but if you wanted to take any notes off of  
6 that, I suppose the time has passed for you to do that.

**MR. BABER:** May I approach? Just one more thing.

**THE COURT:** Go ahead.

9 | BY MR. BABER:

10 Q. Dr. Bloch, I'm going to hand you Trial Exhibit 4027. Just  
11 for completeness you testified earlier and answered some  
12 questions about the Java Language Specification First Edition;  
13 do you recall that?

14 A. Yes, I do.

15 Q. Do you recognize the exhibit I've just put in front of  
16 you?

17 A. Yes. The colors are a little off, but, yes, I recognize  
18 it.

19 Q. What is that exhibit?

20 A. That is the first edition of the Java Language  
21 Specification.

22                   **MR. BABER:** Your Honor, we move admission of the  
23 exhibit.

24 | THE COURT: Number?

25 MR. BABER: 4027.

1                   **THE COURT:** Any objection?

2                   **MR. JACOBS:** No, your Honor.

3                   **THE COURT:** Thank you. Received.

4                   (Trial Exhibit 4027 received

5                   in evidence)

6                   **BY MR. BABER:**

7                   **Q.** Dr. Bloch, I believe you testified earlier that your  
8 recollection was that the First Edition of the Language  
9 Specification included the four core API packages of java.lang,  
10 Java.util, Java.io and Java.net.

11                  I would like you just to look at the index of the  
12 that document and see if that is correct?

13                  **A.** I already saw the packages, but let me make sure I  
14 remembered all of them. If you know which chapter number, tell  
15 me.

16                  **Q.** Chapter 20, 21 and 22 might be helpful.

17                  **A.** Excellent.

18                  All right. So, yes it includes java.lang, java.util  
19 and java.io in their entirety. I do not see java.net, so I'm  
20 going to say java.net is probably not in here.

21                  **Q.** Just quickly, approximately how many pages in the Java  
22 Language Specification are the three chapters dealing with the  
23 API packages, java.lang, java.util and Java.io?

24                  **A.** They go from about 450 to about 750. By my subtraction  
25 that's 300 pages.

1                   **MR. BABER:** Nothing further, your Honor.

2                   **FURTHER REDIRECT EXAMINATION**

3                   **BY MR. JACOBS:**

4                   **Q.** Mr. Bloch, you referred to the resignation of Apache  
5 Software from the JCP; do you recall that?

6                   **A.** I do.

7                   **Q.** Isn't it true that when the Apache Software Foundation  
8 resigned from the Java Community Process, they said that:

9                          "Java specifications are proprietary  
10                         technology that must be licensed directly  
11                         from the spec lead under whatever terms the  
12                         spec lead chooses."

13                   **A.** I don't recall that, but I will take your word for it.

14                   **MR. JACOBS:** Thank you very much, Mr. Bloch.

15                   **THE COURT:** May the witness step down and not be  
16 subject to recall?

17                   **MR. BABER:** He is our witness, your Honor. We may  
18 call him back.

19                   **THE COURT:** All right. How about the plaintiff? Are  
20 you finished with the witness?

21                   **MR. JACOBS:** I believe we are, your Honor.

22                   **THE COURT:** All right. So, Mr. Bloch, you're free to  
23 go. You may be called back by the defendants, but not by the  
24 plaintiffs.

25                          Have a good day. Thank you.

1                   **THE WITNESS:** Thank you.

2                   (Witness excused.)

3                   **THE COURT:** We will now go to the next witness on the  
4 plaintiff's side.

5                   **MR. BOIES:** May I have just a moment, your Honor.

6                   **THE COURT:** Of course.

7                   (Discussion held off the record

8                   amongst counsel.)

9                   **MR. BOIES:** Your Honor, we would next call  
10 Mr. Lindholm.

11                  **THE COURT:** Very well.

12                  All right, welcome. Please raise your right hand.  
13 Stand and raise your right hand.

14                  TIMOTHY LINDHOLM,  
15 called as a witness for the Plaintiff herein, having been first  
16 duly sworn, was examined and testified as follows:

17                  **THE WITNESS:** Yes, I do.

18                  **THE CLERK:** Thank you. Please be seated.

19                  **THE COURT:** All right. The mic will move all around.  
20 You need to be about this close to be heard.

21                  **THE WITNESS:** Sort of like this?

22                  **THE COURT:** Can you all hear okay?

23                  **THE WITNESS:** Close enough?

24                  **THE COURT:** A little closer.

25                  **THE WITNESS:** I'll try about here.

1                   **THE COURT:** Mr. Boies, the floor is yours.

2                   **MR. BOIES:** Thank you. Thank you, your Honor.

3                   **DIRECT EXAMINATION**

4                   **BY MR. BOIES:**

5                   **Q.** Good afternoon, Mr. Lindholm.

6                   **A.** Good afternoon.

7                   **MR. BOIES:** Your Honor, in an effort to move things  
8 along, counsel for Google has properly agreed we can move in  
9 without objection the following exhibits.

10                  **THE COURT:** Go ahead.

11                  **MR. BOIES:** Exhibit 3, 9, 10, 11, 12, 17, 125, 212,  
12 213, 222, 317, 318, 321, 325, and 330.

13                  **THE COURT:** Agreed?

14                  **MS. ANDERSON:** Agreed, your Honor.

15                  **THE COURT:** All right. All of those are in evidence.  
16 Thank you.

17                  (Trial Exhibits 3, 9, 10, 11, 12, 17, 125, 212, 213,  
18 222, 317, 318, 321, 325, and 330 received in  
19 evidence)

20                  **THE COURT:** And for the record, the witness's name  
21 is?

22                  **THE WITNESS:** Timothy Gerald Lindholm.

23                  **THE COURT:** Very well. Welcome again. Go ahead,  
24 counsel.

## **DIRECT EXAMINATION**

BY MR. BOIES:

Q. Mr. Lindholm, you worked at Sun from 1994 to 2005, correct?

**A.** That is correct.

Q. And among your titles there was Distinguished Engineer, correct?

A. That was the last title I held.

9 Q. And you worked on the original Java team which was  
10 involved in the original creation of the Java Platform;  
11 correct, sir?

12 A. That's approximately correct. There was work done on the  
13 Oak language prior to.

14 Q. And, in fact, you have written a couple of books or  
15 coauthored a couple of books, correct?

**A.** Two editions of the same book essentially.

**MR. BOIES:** May I approach the witness, your Honor?

**THE COURT:** You may.

(Whereupon, books were tendered  
to the witness.)

21 | BY MR. BOIES:

22 Q. I'm showing you Exhibits 25 and 987, and I would ask you  
23 if those are the two books?

24 || A. These do appear to be the two books.

25 Q. And could you tell the jury what the title of those two

1 books are?

2 **A.** Well, the first one is called the Java Virtual Machine  
3 Specification, and the second one is the Java Virtual Machine  
4 Specification Second Edition.

5 **Q.** And both of those books you are a coauthor of, correct?

6 **A.** That is correct.

7 **Q.** And both of those books Sun holds the copyright for,  
8 correct?

9                   **MS. ANDERSON:** Objection. Calls for a legal  
10 conclusion, your Honor.

11 **A.** I'm -- I don't recall actually. I would have to look.

12                   **THE COURT:** Well, the -- yes, it does call for a  
13 legal conclusion, but in this -- for this particular question,  
14 it's simple enough that I'm going to let the witness answer the  
15 question.

16                   You can look in the book and tell us whether or not  
17 there is a copyright there?

18                   **THE WITNESS:** There is a copyright notice in the  
19 first one. And it does say "Copyright Sun Microsystems."

20                   And the same is true for the second edition. The  
21 years have changed, but...

22 **BY MR. BOIES:**

23 **Q.** Thank you.

24                   **MR. BOIES:** Your Honor, I would offer Exhibits 25 and  
25 987.

1           **MS. ANDERSON:** No objection, your Honor.

2           **THE COURT:** 25 and 987 received.

3           (Trial Exhibits 25 and 987 received  
4           in evidence)

5           **BY MR. BOIES:**

6           **Q.** Now, while you were at Sun you met with Mr. Rubin; did you  
7 not, sir?

8           **A.** There was at least one -- yes. Yes. There was at least  
9 one occasion that I met with him.

10          **Q.** And Mr. Rubin is now at Google, correct?

11          **A.** That is correct.

12          **Q.** But when you were at Sun and you met with Mr. Rubin,  
13 Mr. Rubin was not yet at Google, correct?

14          **A.** Yes, that's correct.

15          **Q.** He was at a company that was then called Android, correct?

16          **A.** That is correct.

17          **Q.** And that is the company that Google later acquired,  
18 correct?

19          **A.** As I understand it, yes.

20          **Q.** And when you were meeting with Mr. Rubin when he was at  
21 Android, one of the things that you discussed was a possible  
22 agreement between Android and Sun to permit Android to use some  
23 of Sun's intellectual property, correct?

24          **A.** I think that's approximately correct. I don't have  
25 detailed recollections of that meeting.

1 Q. Now, shortly after you met with Mr. Rubin when you were at  
2 Sun and he was at Android, Android was acquired by Google,  
3 correct?

4 A. I don't -- I wouldn't -- I would think that I would say  
5 no. I don't think it was shortly after. I had already -- I  
6 had left Sun. It was some period of time.

7 Q. Well, you -- you left Sun shortly after the time you met  
8 with Mr. Rubin, correct?

9 A. Umm, I don't remember how long it was. It was not a short  
10 while. I think it was a matter of months. Probably quite a  
11 few months.

12 Q. They were both in 2005; is that fair?

13 A. I really don't remember when that meeting was.

14 Q. Okay. Do you remember when you left Sun?

15 A. Yes.

16 Q. When was that?

17 A. Oh, that was -- I remember when I joined Google, which was  
18 July of 2005. I don't remember exactly when I left Sun, but I  
19 think it was just a few weeks prior.

20 Q. And when you joined Google, one of the things that you  
21 worked on was Android; correct, sir?

22 A. Generally speaking, no. I've been -- I have had very  
23 little involvement in Android.

24 Q. Well, all right, sir. Let me ask you to look at Trial  
25 Exhibit 9.

1                   **MR. BOIES:** May I approach, your Honor?

2                   **THE COURT:** You may.

3                   (Whereupon, document was tendered  
4                   to the witness.)

5                   **THE COURT:** It's in evidence. It may be shown to the  
6 jury.

7                   (Document displayed)

8                   **BY MR. BOIES:**

9                   **Q.** And this is a document from you to Mr. Rubin, correct?

10                  **A.** Yes. I think so.

11                  **Q.** And this is October 13th, 2005, correct?

12                  **A.** That's the date that I see here.

13                  **Q.** And this is after you are now both at Google, correct?

14                  **A.** That is correct.

15                  **Q.** And so by this time Google has acquired Android, correct?

16                  **A.** Yes.

17                  **Q.** And if you look at the second paragraph where it says:

18 "Alan." Alan is somebody at Sun, correct?

19                  **A.** I -- I have to think so.

20                  **Q.** And it says:

21                   "Alan presumably wants this both for tactical  
22                   reasons (preserve TCK and implementation  
23                   revenue, defend franchise against  
24                   fragmentation which is his main threat for  
25                   long-term erosion)."

1                   Do you see that?

2 **A.** I see that here.

3 **Q.** And you knew in October of 2005 that one of the things  
4 that Sun was concerned about was possible fragmentation with  
5 respect to Java, correct?

6 **A.** I knew that generally speaking Sun was concerned about  
7 that.

8 **Q.** And you also knew that Sun wanted to preserve their TCK  
9 revenue; correct, sir?

10 **A.** Well, I speculated that here.

11 **Q.** In fact, you didn't say you were speculating it. You  
12 actually said it here; correct, sir?

13 **A.** Well, I think that -- I think I said "Alan presumably  
14 wants." I think I was presuming what was in Alan's head.

15 **Q.** Based in part on your experience at Sun; correct, sir?

16 **A.** That certainly did predate this, yes.

17 **Q.** Now, let me ask you to look at Trial Exhibit 2.

18                   **MR. BOIES:** May I approach, your Honor?

19                   **THE COURT:** You may.

20                   (Whereupon, document was tendered  
21                   to the witness.)

22 **BY MR. BOIES:**

23 **Q.** Now, this is another email that relates to your work on  
24 Android; correct, sir?

25 **A.** I may need to -- I don't recognize this immediately. I

1 may need to scan over this, if I may.

2           **THE COURT:** Is this in evidence?

3           **MR. BOIES:** Yes, Your Honor.

4           **THE COURT:** Are you sure?

5           **THE CLERK:** Yes.

6           **THE COURT:** All right. 2 is in evidence?

7           **THE CLERK:** On the 18th.

8           **THE COURT:** All right.

9 **BY MR. BOIES:**

10          **Q.** This is dated July 26, 2005, correct?

11          **A.** That's what I see here, yes.

12          **Q.** And that's shortly after you joined Google, correct?

13          **A.** That's only like a couple of weeks afterward.

14          **Q.** And this is GPS notes on Android, correct?

15          **A.** That's what the subject of the letter says. I --  
16 apparently, Andy wrote this. I certainly didn't.

17          **Q.** You got a copy of it, though; didn't you, sir?

18          **A.** It appears that I was sent a copy, yes.

19          **Q.** And if you go down to the notes, you'll see three  
20 inferences of things that you said. Where it says "tim,"  
21 "tim," "tim," do you see that?

22          **A.** I do see those lines.

23          **Q.** And those are references to things you said at this GPS  
24 meeting, correct?

25          **A.** Uhm, I don't know that for sure. I don't recall this

1 meeting in any detail. And I didn't write these bullet points  
2 here.

3 **Q.** But you don't have any doubt that you were at this July 26  
4 meeting; do you, sir?

5 **A.** I think I probably was at it. I'm just saying I don't  
6 remember it.

7 **Q.** Well, right above the references to you there's something  
8 that Brian said. Do you see that?

9 **A.** Yes.

10 **Q.** Who is Brian?

11 **A.** I'm not sure I know.

12 **Q.** Brian says -- is asserted to have said:

13                   "Concerned about Sun preventing distributing  
14                   Java as open source."

15                   Do you see that?

16 **A.** I do see that.

17 **Q.** Did you understand in July of 2005, that Sun had the right  
18 and ability to prevent distributing Java as open source?

19                   **MS. ANDERSON:** Objection. Foundation. Legal  
20 opinion, as well.

21                   **THE COURT:** Well, he might have had that in his mind  
22 as an issue even if -- it doesn't necessarily call for a legal  
23 opinion at this time. The question calls for what was in his  
24 mind, as a historical matter, back then in 2005. So the  
25 objection is overruled.

1           So the question is, did you have that point in mind  
2 at some point about the time of this meeting? That's what  
3 counsel is asking.

4           **THE WITNESS:** Uhm, I don't recall that I had this --  
5 this point in my mind at the time of this meeting. And it  
6 doesn't seem to be attributed to me here.

7           **THE COURT:** Well, maybe I didn't -- counsel was  
8 asking, was that discussed in your presence back then?

9           **THE WITNESS:** I don't recall it to have been  
10 discussed. It could have been, but I don't recall.

11           **THE COURT:** All right. Thank you. Go ahead,  
12 counsel.

13           **MR. BOIES:** May I approach, Your Honor?

14           **THE COURT:** Yes.

15 **BY MR. BOIES:**

16 **Q.** Let me show you trial Exhibit 3, sir. And this is a  
17 document that you received, correct?

18 **A.** Let me glance at this. It's multipage.

19 **Q.** Right on the first page you see "From: Andy Rubin," "To:  
20 Tim Lindholm"?

21 **A.** I see that.

22 **Q.** (As read:)

23           "Tim, the enclosed document starts to  
24 memorialize our thinking."

25           Do you see that?

1     **A.** Yes, I do see that.

2     **Q.** And then if you go to the next document, the next page, it  
3 talks about "Licensing Discussions with Sun."

4                  Do you see that?

5     **A.** I do -- I see that as the title of this page, if that's  
6 what you're referring to.

7     **Q.** Yes. And you were involved in licensing discussions with  
8 Sun; were you not, sir?

9     **A.** Uhm, I had a couple of meetings with Sun. It was much  
10 later, I think, than this document.

11    **Q.** Let's go on to the next page, where it says,  
12 "Requirements." Do you see the first requirement there says,  
13 "Google needs a TCK license"?

14    **A.** I see that bullet point, yes.

15    **Q.** And was that your understanding in 2005, sir?

16    **A.** Uhm, that was not my general understanding.

17    **Q.** Was that your specific understanding?

18    **A.** Well, if -- looking at this, this seems to be a discussion  
19 of CLDC licensing with Sun. And if Google were to license the  
20 CLDC from Sun, then I think that that would mean a TCK license.  
21 But there is -- this is a very specific structure that's being  
22 discussed here.

23    **Q.** This was a specific discussion with respect to Android,  
24 correct?

25    **A.** Uhm, I -- I -- I really haven't -- don't recall having

1 seen this in anytime -- you know, I may have seen it when it  
2 was first put out, but I don't recall any discussion or  
3 anything about it.

4 Q. Well, let's just look at the very next bullet.

5 A. Okay.

6 Q. Where it says:

7 "We propose the Android product must pass the  
8 TCK."

9 Do you see that?

10 A. I do see that.

11 Q. It also, at the end, says:

12 "OEM licensees must pass TCK again on  
13 shipping product."

14 Do you see that?

15 A. I do see that.

16 Q. Was that your understanding at the time?

17 A. That was my understanding of how it worked if you got a  
18 CLDC license from Sun. But that's -- this is predicated on --  
19 on having -- on what you do if you're going to get a CLDC  
20 license with Sun.

21 Q. And you were discussing a couple of different approaches  
22 with Sun; correct, sir?

23 A. Uhm, at this time, I wasn't discussing anything with Sun.

24 Q. Well, let me --

25 MR. BOIES: May I approach, Your Honor?

1 THE COURT: Yes.

2 BY MR. BOIES:

3 Q. Look at Trial Exhibit 12.

4 A. Okay.

5 Q. This is an e-mail exchange between you and Mr. Lindholm,  
6 correct?

7 A. Uhm, between me and Mr. Rubin; do you mean?

8 Q. Between you and Mr. Rubin, excuse me, yes.

9 A. Uhm, yes. That seems to be correct.

10 Q. In December of 2005?

11 A. Yes.

12 Q. And Mr. Rubin writes:

13 "My reasoning is either a) we'll partner with  
14 Sun as contemplated in our recent  
15 discussions, or b) we'll take a license."

16 Do you see that?

17 A. I do see that.

18 Q. Was that your understanding in December of 2005?

19 A. No, I don't think -- I don't think it was my  
20 understanding. It seems to be what Andy wrote, not my  
21 understanding.

22 Q. Did you ever tell Mr. Rubin that that was not your  
23 understanding?

24 A. Uhm, I -- I can't remember not telling him that, but that  
25 isn't, I don't think, the sort of thing I would remember.

1                   **MR. BOIES:** Let me ask you -- may I approach, Your  
2 Honor?

3                   **THE COURT:** Yes.

4                   **BY MR. BOIES:**

5                   **Q.** Let me ask you to look at Trial Exhibit 326.

6                   **MR. BOIES:** Which is not in evidence, so don't  
7 display it.

8                   **BY MR. BOIES:**

9                   **Q.** This is an e-mail from Bob Lee to you, dated February 20,  
10 2009, correct?

11                  **A.** That's what it appears to be.

12                  **Q.** And you, in turn, had written him earlier that same day,  
13 correct?

14                  **A.** Yes. As I read -- read backwards through the e-mail, yes.

15                  **Q.** And prior to that, you had gotten an e-mail from Dave  
16 Sobota, correct?

17                  **A.** I can't tell from this whether I had received that or --  
18 these e-mails are nested kind of improperly. So I can't  
19 actually tell what -- how these arrived in this particular  
20 sequence.

21                  **Q.** Well, let's go on to the next page, just to try to clarify  
22 that.

23                   Do you see right under the word "Confidential" where  
24 it says, "Hi Tim, Bob and Jeremy"?

25                  **A.** Yes, I do see that.

1 Q. Tim is you, correct, sir?

2 A. I -- I -- I have to assume so, but I can't -- I can't  
3 verify this from this document.

4 MR. BOIES: Your Honor, I would offer Exhibit 326.

5 MS. ANDERSON: Your Honor, we object on both Rule 403  
6 and foundation grounds.

7 THE COURT: All right. Overruled. 326 is received.

8 (Trial Exhibit 326 received in evidence.)

9 BY MR. BOIES:

10 Q. Now, let me ask you to look at the next page of the  
11 document.

12 A. Is this the final page?

13 Q. The final page. And they're talking about a possible deal  
14 with Sun, correct, here?

15 A. Where do you see that?

16 Q. Where it says "Hi, Tim."

17 A. Yes.

18 Q. And they then go on to describe a possible deal between  
19 Google and Sun; correct, sir?

20 A. Uhm, that seems to be what the -- what the author of this  
21 attached e-mail was reporting about some other guy's thoughts.

22 Q. And reporting to you, correct, sir?

23 A. Uhm, I have to infer that they were passing these on to me  
24 as well as others.

25 Q. Now, if we go to the last page, it says:

1                   "This deal would be good for Google."

2                   Do you see that?

3 **A.** I do see that.

4 **Q.** And the first reason it will be good for Google is, "Our  
5 Java lawsuits go away." Do you see that, sir?

6 **A.** I do see that.

7 **Q.** And you didn't have any Java lawsuits, actually, at that  
8 time; did you?

9 **A.** Not that I'm aware of.

10 **Q.** What you were concerned about were the Java lawsuits that  
11 would come if you did not get a license for Sun; correct, sir?

12 **A.** I wasn't concerned about that, and I didn't write this.  
13 So, you know, I did not have that concern.

14 **Q.** Did you ever tell this person that you didn't have that  
15 concern, that his concern was not reasonable?

16 **A.** I don't recall whether I did or didn't. I don't recall,  
17 really, what came of this. This was -- I don't recall what  
18 came of this kind of flier from this person.

19 **Q.** Let me ask you to look at a document that you did write,  
20 Trial Exhibit 10.

21                   **MR. BOIES:** May I approach, Your Honor?

22                   **THE COURT:** Yes.

23 **BY MR. BOIES:**

24 **Q.** Now, this is a document that you wrote August 6th, 2010,  
25 correct?

1 A. That's correct.

2 Q. And by this time, Android is already shipping, correct?

3 A. Yes, I think it shipped multiple versions, at this  
4 pointed.

5 Q. And if you go to the second paragraph here --

6 A. Yes.

7 Q. This is an e-mail from you to Andy Rubin, correct?

8 A. Among others, yes.

9 Q. Among others. And you say:

10 "What we've actually been asked to do by  
11 Larry and Sergey ..."

12 Now, "Larry" there is Larry Page, correct?

13 A. Yes.

14 Q. And who is "Sergey"?

15 A. Sergey Brin, the other Google founder.

16 Q. These are the two Google founders, correct?

17 A. That is correct.

18 Q. And these are two of the top three executives at Google,  
19 correct?

20 A. That's my understanding of the corporate structure of  
21 Google, yes.

22 **MR. BOIES:** Excuse me. Could we publish it? I'm  
23 sorry.

24 **THE COURT:** Not yet in evidence.

25 **MR. BOIES:** We offered this at the beginning, Your

1 Honor.

2           **THE COURT:** Oh, yes, you did. It's in evidence. It  
3 should be published.

4           **MR. BOIES:** I apologize.

5           (Document displayed.)

6 **BY MR. BOIES:**

7   **Q.** The second paragraph.

8   **A.** Okay.

9   **Q.** (As read:)

10           "What we've been asked to do by Larry and  
11           Sergey..."

12           And those are two of the top three executives at  
13 Google, correct?

14   **A.** That is correct.

15   **Q.** (As read:)

16           "... is to investigate what technical  
17           alternatives exist to Java for Android and  
18           Chrome."

19           Do you see that?

20   **A.** I do.

21   **Q.** And then you say:

22           "We've been over a bunch of these, and think  
23           they all suck."

24           Do you see that?

25   **A.** I do, indeed.

1 Q. And then you say:

2 "We conclude that we need to negotiate a  
3 license for Java under the terms we need."

4 Do you see that?

5 A. I do see that.

6 Q. And that was a license from Sun; correct, sir?

7 **MS. ANDERSON:** Objection, Your Honor. Motion in  
8 limine.

9 **THE COURT:** I don't remember that.

10 **MS. ANDERSON:** With regard to Oracle's Motion in  
11 Limine No. 5, Your Honor. Your Honor placed limitations on  
12 questions regarding the language.

13 **THE COURT:** Well, I probably did, but it's not coming  
14 back to me.

15 Counsel, do you know what counsel is talking about?

16 **MR. BOIES:** I do -- I think I do, Your Honor, but I  
17 don't think it would apply to simply asking whether this  
18 license that he refers to was a license from Sun.

19 **THE COURT:** I will allow the question.

20 **MS. ANDERSON:** Thank you, Your Honor.

21 **THE COURT:** Overruled. Go ahead.

22 **BY MR. BOIES:**

23 Q. That's just a yes or no question.

24 A. Could you repeat the question.

25 Q. Sure. When you refer to a license for Java here, that's a

1 license from Sun, correct?

2 **A.** That is not what I meant in this e-mail.

3 **Q.** Who was the license from?

4 **A.** I would -- what I actually meant --

5 **Q.** Sir --

6 **A.** Excuse me?

7 **Q.** I'm asking you --

8                   **MR. BOIES:** And I would ask the judge to instruct the  
9 witness to try to be directly responsive to this question  
10 because we will get into the motion in limine otherwise.

11                  **THE COURT:** All right. Say precisely your question.

12                  **MR. BOIES:** Okay.

13 **BY MR. BOIES:**

14 **Q.** And I'll first ask the predicate question.

15                  When you referred to a license here, was it a license  
16 from Sun?

17 **A.** No, it was not.

18 **Q.** Was it a license from some other company?

19 **A.** No, it was not.

20 **Q.** Was it a license from anybody?

21 **A.** It was not specifically a license from anybody.

22                  **MR. BOIES:** Thank you, Your Honor. I have no more  
23 questions.

24                  **THE COURT:** All right. Any more questions?

25                  **MS. ANDERSON:** Very briefly, Your Honor.

## CROSS EXAMINATION

**BY MS. ANDERSON:**

Q. We're now good afternoon, Mr. Lindholm.

**A.** Been a long day already.

Q. Very briefly, would you tell the jury where you live.

A. I live in Palo Alto, California.

Q. And how long have you lived in Northern California?

**A.** I've lived here since about 1985. In Palo Alto, for 21 years.

**Q.** And what is your educational background?

11 A. I grew up in a tiny town in northern Minnesota. Went to  
12 public school there, and then went to a liberal arts college in  
13 southern Minnesota. Got a bachelor's degree.

14 Q. So we know you work at Google today, from your testimony  
15 earlier.

When did you start working at Google?

17 A. Uhm, I started working July 7th of 2005, the first day  
18 after the Fourth of July holiday.

19 Q. You mentioned before that you haven't spent much time  
20 working on Android. What has been your job during the years  
21 you've worked at Google?

22 A. The whole time I've been at Google, I've worked in a piece  
23 of the organization called infrastructure, which is the  
24 organization that makes Google's big data centers work  
25 properly. So things like Gmail and Web Search run on these --

1 in data centers. And we have to make many thousands of  
2 computers work sort of as one to create these services.

3 **Q.** And, briefly, during your years at Sun, what generally  
4 were your job responsibilities?

5 **A.** My -- my job responsibilities have been entirely around --  
6 around the infrastructure project. I have specifically focused  
7 on what we call utilization, trying to get the most use out of  
8 the computers that we have available.

9           Because if you can get more use out of them, you  
10 don't have to buy as many.

11 **Q.** And please remind the jury, what were the years that you  
12 were with Sun?

13 **A.** I was with Sun from, I think, March of 1994 until, I  
14 believe, June or July of 2005.

15 **Q.** Counsel previously showed you some treatises that you  
16 authored related to Java. During your years with Sun, did you  
17 have experience with the Java platform?

18 **A.** Yes, I did.

19 **Q.** Okay. And did you gain an understanding during those  
20 years about the concept of Java API packages?

21 **A.** Oh, yes, I did.

22 **Q.** Did you become generally familiar with how they're  
23 organized?

24 **A.** Yes, absolutely.

25 **Q.** And as a general matter how, generally, do you understand

1 the fashion or organization of APIs?

2 **A.** I understand the organization quite well. I didn't  
3 necessarily understand the details of every API equally well.

4 **Q.** Certainly.

5 Based on your experience over the years understanding  
6 Java and the platform, have you had a view as to whether or not  
7 API packages and their organization, the organization of API  
8 packages, is freely available for use?

9 **MR. BOIES:** Objection, Your Honor. Scope and  
10 foundation.

11 **MS. ANDERSON:** Your Honor, counsel asked questions  
12 about copyrights and a treatise entirely devoted to Java.

13 **THE COURT:** It's outside the scope. This particular  
14 question is outside the scope.

15 **MS. ANDERSON:** Thank you, Your Honor.

16 **THE COURT:** Now, that's without premise to you  
17 recalling the witness in your case and possibly answering the  
18 same question. I'm not ruling on any other objection.

19 **MS. ANDERSON:** Your Honor mentioned earlier -- this  
20 would be very brief, literally a minute or two, to avoid having  
21 the witness have to return.

22 **THE COURT:** Any objection to going outside the scope?

23 **MR. BOIES:** Yes, Your Honor.

24 **THE COURT:** I'm going to allow it. How many  
25 questions do you have?

1                   **MS. ANDERSON:** It looks like three.

2                   **THE COURT:** All right. Go ahead. Is this the only  
3 three questions outside the scope?

4                   **MS. ANDERSON:** Yes. That's my understanding, Your  
5 Honor.

6                   **THE COURT:** Then you may ask these questions.

7                   **MS. ANDERSON:** Thank you.

8 **BY MS. ANDERSON:**

9                   **Q.** Based on your experience with Java over the years, have  
10 you held a view over whether or not the organization of API  
11 packages is something that has been freely available for use?  
12 And I'm referring to Java API packages.

13                   **MR. BOIES:** Objection. Foundation. Calls for a  
14 legal conclusion, as well.

15                   **THE COURT:** Well, the jury will understand we don't  
16 have a lawyer up here. And this will just be a lay opinion.  
17 It's not a legal opinion. But I'm going to allow the question  
18 with that caveat.

19                   Please answer.

20                   **THE WITNESS:** Okay. Uhm, as a software engineer, not  
21 a lawyer, it's -- it's always been my understanding that --  
22 that the organization of software APIs are free for use by  
23 other people.

24 **BY MS. ANDERSON:**

25                   **Q.** Thank you.

1                  You testified earlier about some communications that  
2 you had with Sun representatives while you were with Google.

3 Do you remember that testimony?

4 **A.** Which ones are you referring to?

5 **Q.** Certainly. Let's back up for a moment. Right before you  
6 joined Google --

7 **A.** Correct.

8 **Q.** -- before you were brought on as an actual employee, did  
9 you have any knowledge or information about whether Google was  
10 about to acquire Android?

11 **A.** I had no knowledge, whatsoever. I was completely  
12 surprised.

13 **Q.** And you were asked questions about some e-mails that  
14 reflected discussions on the subject of Android. Do you recall  
15 that?

16 **A.** Yes. But I -- I'm not sure if you're referring to  
17 specific ones.

18 **Q.** Thank you.

19                  Did you do any work at all related to the subject of  
20 Android during your years at Google?

21 **A.** No, I wasn't part of the design, the implementation, the  
22 marketing. I really played no substantial role in the Android  
23 technology.

24 **Q.** Right. And I'd like to bring the question out a little  
25 broader. I'm not specifically limiting to the Android

1 technology itself, but as a broader matter have you done  
2 anything to assist the Android project at Google in any way  
3 over the years?

4 **A.** I've occasionally been asked by Android project members  
5 who were part of the project for advice on their project.

6 **Q.** Thank you.

7 During the years that you have been at Google, have  
8 you ever contributed any source code, whatsoever, to the  
9 Android project?

10 **A.** Not a line. Zero.

11 **Q.** Have you wrote any specification, whatever, for the  
12 Android platform?

13 **A.** Not a bit.

14 **Q.** Did you participate personally in making decisions about  
15 the architecture of Android?

16 **A.** No. Happened completely outside of my vision.

17 **Q.** All right. Among the exhibits that have been admitted by  
18 counsel for Oracle, reflect some comments by you regarding the  
19 notion of a license being critical.

20 I would like to show you Trial Exhibit 17, which  
21 should be before you there. Do you have that?

22 **A.** I don't think I do have that here.

23 **Q.** Okay. All right.

24 **MS. ANDERSON:** Can I ask, Gary, Trial Exhibit 17?

25 **THE COURT:** It's been moved into evidence, so you can

1 just put it on the screen, if you wish.

2 (Document displayed.)

3 **THE WITNESS:** There we go.

4 **MS. ANDERSON:** May I approach, Your Honor?

5 **BY MS. ANDERSON:**

6 **Q.** Please take a look at Trial Exhibit 17.

7 **MS. ANDERSON:** Is it available for the jury, as well?

8 Thank you.

9 **BY MS. ANDERSON:**

10 **Q.** Mr. Lindholm, do you recognize this e-mail exchange?

11 **A.** Yes, I do.

12 **Q.** All right. And I'd like to draw your attention to that  
13 first paragraph there, of this e-mail.

14 **A.** Uh-huh.

15 **Q.** And in this e-mail you wrote to Mr. Cockburn. Do you see  
16 that?

17 **A.** Yes. "Hi, Bill."

18 **Q.** In that second sentence of the first paragraph of this  
19 e-mail you talk about "helping negotiate with an old team at  
20 Sun for a critical license."

21 Do you see that reference?

22 **A.** Yes, I do.

23 **Q.** All right. Please tell the jury, just as a general  
24 matter, what was the purpose of your sending this e-mail?

25 **A.** The purpose of sending this e-mail was, I worked for Bill

1 Coran (phonetic) and he, as far as I know, wasn't aware that I  
2 was talking -- that I had talked with anybody in the Android  
3 camp.

4 I had now been asked to go on a trip that would take  
5 me away from my normal duties, and so I felt the need to kind  
6 of say, hey, well, Bill, I've been doing this thing. As a  
7 result, I've been asked to go on this trip. Is that okay for  
8 me to take time off of my normal job to go on this trip?

9 **Q.** When you used the word "critical" what did you mean?

10 **A.** So --

11 **MR. BOIES:** Objection, Your Honor.

12 **THE COURT:** What's the objection?

13 **MR. BOIES:** There's no foundation for his  
14 interpreting this document years after the fact.

15 **THE COURT:** He wrote it.

16 You can rephrase the question and say: Why was it  
17 critical? Why did he think it was critical at the time. "At  
18 the time" would be a proper question. Why did he think it was  
19 critical. That's a proper question.

20 Go ahead.

21 **MS. ANDERSON:** Be happy to, Your Honor.

22 **BY MS. ANDERSON:**

23 **Q.** Mr. Lindholm, at the time that you wrote this e-mail, why  
24 did you write that you believed it was a critical license?

25 **A.** I think that there are two reasons --

1                   **THE COURT:** No, no. Not "I think now."

2                   **THE WITNESS:** Oh, okay.

3                   **THE COURT:** It's what you thought back then. There's  
4 a difference. One is after the fact, and one was at the time.  
5 We don't want after the fact. We want at the time. What did  
6 you think?

7                   **THE WITNESS:** Thank you, Your Honor.

8                   At the time, what I wanted to do was to tell my boss  
9 that this was important. What I was doing was an important  
10 thing. I also, at that point, may -- well, I also suspect that  
11 at this point I used the word --

12                  **THE COURT:** No, no.

13                  **THE WITNESS:** Okay.

14                  **THE COURT:** We don't go on suspicions.

15                  **THE WITNESS:** Okay.

16                  **THE COURT:** We go on actual memory. So no more --

17                  **THE WITNESS:** Okay.

18                  **THE COURT:** The answer is over. Next question.

19                  **BY MS. ANDERSON:**

20                  **Q.** Are there any other reasons that you know today that back  
21 at the time you wrote the e-mail you characterized --

22                  **THE COURT:** Calling for speculation. It's what his  
23 reasons were back then that matters, and that he can actually  
24 recall, not some explanation today.

25                  **MS. ANDERSON:** Understood, Your Honor.

1                   **THE COURT:** There's a difference. We want to stick  
2 with what was in his mind back then. Okay.

3                   **MS. ANDERSON:** Thank you.

4                   **THE COURT:** You can try again.

5 **BY MS. ANDERSON:**

6 **Q.** Are there any further reasons that were in your mind back  
7 then when you wrote this e-mail?

8 **A.** At the time I wrote this e-mail, Google and Sun had been  
9 discussing a -- a co-development agreement that -- that would  
10 have involved Google using Sun's source code as the basis for  
11 some of the Android work.

12                  At that time, a license -- well, in that agreement, a  
13 license would have been necessary because we were going to use  
14 Sun source code.

15 **Q.** Thank you.

16                  **MS. ANDERSON:** Your Honor, may I approach again?

17                  **THE COURT:** Yes.

18 **BY MS. ANDERSON:**

19 **Q.** Mr. Lindholm, I'm placing before you two exhibits. And  
20 I'll take Exhibit 318 first. If you would have that open.

21 **A.** Yes. Okay. I have it open now.

22                  **MS. ANDERSON:** This has been admitted, Your Honor.

23                  (Document displayed.)

24 **BY MS. ANDERSON:**

25 **Q.** What is Exhibit 318, Mr. Lindholm?

1     **A.** It is a set of -- it is a set of e-mail notes that I sent  
2 to myself, notes to myself sent as an e-mail to myself.

3     **Q.** And the subject of your e-mail?

4     **A.** "Android notes."

5     **Q.** Drawing your attention to the very last sentence of this  
6 e-mail.

7     **A.** Yes.

8     **Q.** Where you wrote, "I'm going to get in all possible loops  
9 around this project," what did you mean back then when you  
10 wrote it?

11    **A.** Back then, I -- I had -- I had just been surprised by this  
12 new project which I had had no expectation of. And I was -- I  
13 was at a point in my time at Google where I was told, figure  
14 out a project to work on. You know, take a month and decide  
15 what you want to work on.

16               So this was -- this e-mail was sent to myself one  
17 week after I joined. And I -- and I -- what I was saying there  
18 was I should get involved in this. This is -- this is a cool  
19 new project that I should try to get involved in.

20    **Q.** And did that end up happening? Did you end up getting in  
21 all possible loops around the Android project?

22    **A.** No. I was involved for about a year, at the beginning, in  
23 the discussion of this co-development agreement I had mentioned  
24 earlier. But after that happened, I had very, very little --  
25 after that co-development agreement did not go through, I had

1 very, very little interaction with Android after that point.

2 **Q.** And, again, I'd like to draw your attention to Exhibit  
3 321, which I had also just placed before you.

4 Do you have that before you, Mr. Lindholm?

5 **A.** I do.

6 **Q.** And this is -- is this an e-mail exchange that you  
7 participated in with Mr. Rubin?

8 **A.** Yes, I think so.

9 **Q.** All right. And please tell the jury, what's the date of  
10 the e-mail exchange?

11 **A.** The date is August 9th, 2005.

12 **Q.** And approximately how long was this after you had joined  
13 Google?

14 **A.** That was about -- about a month.

15 **Q.** All right. And drawing your attention to the --  
16 basically, the middle of the page, in the middle of that  
17 largest paragraph, with the sentence that starts, "I think my  
18 main value." Do you see that sentence?

19 **A.** Yes, I do.

20 **MS. ANDERSON:** All right. Can we get that  
21 highlighted, please, Ben, "I think my main value would be."  
22 Thank you. There you go.

23 **BY MS. ANDERSON:**

24 **Q.** So in this particular sentence, you describe a potential  
25 role. Could you please tell the jury what the role was you

1 were thinking at the time and whether you, in fact, played that  
2 role at Google?

3 **A.** Yes. I was thinking -- well, my value as a J2ME runtime  
4 generalist, that involves writing a virtual machine for --  
5 potentially for the Android platform. The sort of thing that I  
6 had done in the 1990s, at Sun. That was one possible role.

7 Another role would be to help -- help Android  
8 understand the ecosystem around platforms like this, because my  
9 experience at Sun was largely focused not even necessarily so  
10 much around coding, but around creating this Java ecosystem  
11 that han sprung up in the 1990s and 2000s.

12 **Q.** Did you ever end up playing the role that you talk about  
13 in that sentence, while at Google?

14 **A.** No, I never did either of these things for Android.

15 **Q.** Would you please pull out, Mr. Lindholm, Exhibit 326,  
16 which you were shown earlier by Oracle's counsel.

17 (Document displayed.)

18 **A.** I have that out now.

19 **Q.** Thank you. First of all, as you sit here today, do you  
20 ever recall receiving this e-mail back in February of 2009?

21 **A.** I think I do recall receiving it.

22 **Q.** All right. Do you know who Dave Sobota is, whose name  
23 appears as one of the authors within this e-mail exchange?

24 **A.** No, I don't believe I know Dave.

25 **Q.** All right. And then drawing your attention in this

1 document down to the second page, you see one of the e-mail --  
2 e-mails that was authored by Mr. Sobota. And underneath it it  
3 says, "Confidential. Hi, Tom, Bob, and Jeremy."

4 **A.** "Tim, Bob, and Jeremy."

5 **Q.** "Tim, Bob, and Jeremy."

6 **A.** Yes.

7 **Q.** Excuse me. And then in the first sentence it says, "Josh  
8 Bloch, Brett Slatkin and I." Do you see that?

9 **A.** I do.

10 **Q.** Do you know who Brett Slatkin is?

11 **A.** I recognize his name, but I don't know him.

12 **Q.** And below this, in the next paragraph, do you see where it  
13 says "Brett's thoughts"?

14 **A.** Yes.

15 **Q.** And you were asked some questions by Oracle's counsel  
16 about this discussion. Do you recall that?

17 **A.** I recall just a while ago, yes.

18 **Q.** All right. With regard to this discussion of a potential  
19 arrangement for purchasing something from Sun, did this ever  
20 occur?

21 **A.** No, definitely not. Or at least as far as I know.

22 **Q.** Did you ever observe that anything came of this proposal  
23 that's supposedly from Mr. Slatkin?

24 **A.** I don't know of anything that came from this.

25 **Q.** All right. Specifically, drawing your attention to the

1 section that says, "Good for Google," which is on the third  
2 page of this exhibit, where it says, "Our Java lawsuits go  
3 away."

4 **A.** I see that, yes.

5 **Q.** What was your reaction to reading a sentence about Google  
6 lawsuits going away?

7 **A.** I -- I didn't know what that meant because, as far as I  
8 know, there weren't any Java lawsuits.

9 **Q.** And, specifically, at the time were you aware of any  
10 lawsuits against Google related in any way to Android, at the  
11 time that you received this e-mail exchange?

12 **A.** No, I don't think so.

13 **Q.** All right. I'd like to draw your attention to Exhibit  
14 330.

15 **MS. ANDERSON:** Which is also already in evidence Your  
16 Honor.

17 (Document displayed.)

18 **THE WITNESS:** Do I have that? I don't think I  
19 have --

20 **MS. ANDERSON:** You don't have 330 before you? Let me  
21 put it before you. I have it here. Excuse me.

22 Your Honor, may approach?

23 **THE WITNESS:** Thank you.

24 **BY MS. ANDERSON:**

25 **Q.** Thank you.

1     **A.**    Okay.

2     **Q.**    With regard to Exhibit 330, drawing your attention to this  
3 first paragraph of this e-mail exchange --

4     **A.**    Yes.

5     **Q.**    -- first of all, is this an e-mail that you wrote at least  
6 the first part of the string?

7     **A.**    Yes, it is.

8     **Q.**    All right. And you see here there's a comment about  
9 Microsoft within this e-mail. Do you see that?

10    **A.**    Yes, I see this.

11    **Q.**    And within your e-mail you talk about Java in relation to  
12 Sun and in relation to Microsoft and some languages. Do you  
13 see that?

14    **A.**    Yes, I do.

15    **Q.**    All right. There's a sentence within that paragraph where  
16 you say:

17                "Of course, if they had just done C# and .Net  
18                at the time, rather than trying to steal  
19                Java, they might well have won."

20                Do you see that sentence?

21    **A.**    I do.

22    **Q.**    Now that you're oriented, first of all, who are you  
23 referring to as trying to steal Java?

24    **A.**    Microsoft, I think.

25    **Q.**    What did you mean by that comment?

1     **A.** This was relating back to the litigation that Sun and  
2 Microsoft nearly entered into back in the 1990s.

3     **Q.** Without going into any great detail here, my question to  
4 you is, what, if anything, did you understand Microsoft had  
5 done to try to steal Java, according to this e-mail?

6     **A.** According -- could you rephrase that, please.

7     **Q.** Sure. I'm trying to track along the e-mail language here.  
8 You make a reference to stealing of Java.

9                  What is it that you thought Microsoft had done, which  
10 was trying to steal Java?

11    **A.** I thought that they took -- they took -- they manipulated  
12 their Visual J++ product so as to be a modest difference  
13 from -- a modest change from Sun's standard Java in such a way  
14 that people who would program to their Visual J++ product would  
15 have their applications locked to it.

16                  It would -- the point of Java was to be able to -- an  
17 application should be able to work on multiple implementations.  
18 But Microsoft was manipulating the situation such that if you  
19 wrote to their Visual J++ product, your application might no  
20 longer work on Sun's product or IBM's Java product or the  
21 various industries' products.

22    **Q.** And how about how it was branded?

23    **A.** Microsoft Visual J++ was initially branded with the cup  
24 and steam and the Java -- the Java brand, and full -- full  
25 statements of compatibility and everything.

1 Q. Thank you.

2 THE COURT: You're almost out of time.

3 MS. ANDERSON: I have a few more questions, and I'll  
4 be finished, Your Honor. If that's okay with Your Honor.

5 THE COURT: At 1:00 o'clock we'll stop. If Mr. Boies  
6 has more questions, the witness will have to stay over.

7 BY MS. ANDERSON:

8 Q. Drawing your attention to Exhibit 10. I don't actually  
9 have specific questions for you about this e-mail, but want to  
10 orient you to the time period, which is August of 2010.

11 A. Yes.

12 Q. In August of 2010, had you reviewed Android's source code?

13 A. No --

14 MR. BOIES: Objection, Your Honor. This is the  
15 motion in limine.

16 THE COURT: Just a moment.

17 Was this question asked at the deposition?

18 MS. ANDERSON: Not at all, Your Honor.

19 MR. BOIES: You have the in limine motion.

20 THE COURT: I have the order. Show me the question  
21 in the deposition where it comes close to this one.

22 MR. BOIES: Uhm, the point -- the point I was relying  
23 in the order was where it says, "First" and what follows right  
24 from there. I can try to find the deposition.

25 THE COURT: Well, but the -- what was your question,

1 whether he read -- whether he reviewed the which?

2 **MS. ANDERSON:** As of August of 2010, had Mr. Lindholm  
3 ever reviewed Android source code.

4 **THE COURT:** Android?

5 **MS. ANDERSON:** Yes.

6 **THE COURT:** I do think that that is encompassed  
7 within the motion in limine. So that objection is sustained.

8 **MS. ANDERSON:** Okay. Thank you. No further  
9 questions, Your Honor.

10 **THE COURT:** All right. Any follow-up?

11 **MR. BOIES:** Your Honor, do you have time for one  
12 question?

13 **THE COURT:** One question.

14 **MR. BOIES:** One question.

15 May I approach, Your Honor? Perhaps we can just put  
16 up on the screen Trial Exhibit 12, which is one of the ones  
17 that we offered in evidence.

18 (Document displayed.)

19 **REDIRECT EXAMINATION**

20 **BY MR. BOIES:**

21 **Q.** You told counsel a few minutes ago that the reason that  
22 you thought that Google needed a license was because they were  
23 discussing a partnership agreement.

24 Do you recall that?

25 **A.** Uhm, this is -- yes, yes, with reference to the 2006 era.

1 Q. Right. Now, here, if you look at what Mr. Rubin wrote  
2 you, "My reasoning... "

3 A. "My reasoning, yes.

4 Q. Is that:

5 "... either a) we'll partner with Sun as  
6 contemplated in our recent discussions, or b)  
7 we'll take a license."

8 Do you see that?

9 A. I do see that.

10 Q. So what Mr. Rubin was saying was that they would need a  
11 license if they did not partner with Sun, correct? That was  
12 the alternative?

13 A. Uhm, I -- I can only understand what Mr. Rubin wrote here.  
14 He gives these two alternatives. I don't know his reasoning  
15 behind that. He says it's his reasoning but I don't know what  
16 his reasoning was.

17 Q. You do know Mr. Rubin was responsible for Android?

18 A. Yes, I do.

19 **MR. BOIES:** No more questions.

20 **MS. ANDERSON:** No further questions, Your Honor.

21 **THE COURT:** May the witness now be excused except --  
22 you said you weren't going to recall him anyway, right?

23 **MS. ANDERSON:** That is my understanding, is, we're  
24 not recalling him. That's correct, Your Honor.

25 **THE COURT:** May he be excused?

1                   **MR. BOIES:** Yes, Your Honor.

2                   **MR. JACOBS:** Yes, Your Honor.

3                   **THE COURT:** You are free from any subpoena. You are  
4 free to go. Leave those documents here. We'll take care of  
5 them.

6                   **THE WITNESS:** Okay.

7                   **THE COURT:** Over there in the jury box, remember the  
8 admonition. We'll see you back here at 7:45. And we're making  
9 good progress. If you continue to be here on time, I think  
10 we'll -- you know, we're doing good on the time. So thank you.  
11 Have a good evening.

12                  **THE CLERK:** All rise.

13                  (Jury out at 1:01 p.m.)

14                  **THE COURT:** Mr. Lindholm, you may step down. Thank  
15 you.

16                  **THE WITNESS:** Thank you.

17                  **THE COURT:** You're excused. You're an ordinary  
18 civilian again.

19                  (Witness gestures.)

20                  (Laughter)

21                  **THE COURT:** You can stay in the courtroom. Thank  
22 you.

23                  One moment.

24                  Time so far, 435 minutes out of a total of 1,020  
25 allocated. Plaintiff has used 435. Defendant has used 256.

1 All right. It's right up-to-date.

2 Any issues before we break for the day? Yes, sir.

3 **MR. JACOBS:** Two things with respect to upcoming  
4 expert testimony, Your Honor. Let's make it three so we have  
5 the lay of the land.

6 Issue number one: The way the expert reports came in  
7 there was the opening, there was the responses, there was  
8 rebuttal expert reports.

9 As the issues have evolved in the case, that which is  
10 in rebuttal in some cases is more pertinent than that which is  
11 in opening. Your ground rules, however, as written, are pretty  
12 clear that the expert can only testify to what's in the opening  
13 report on our opening case. I think that's going to lead, in  
14 this particular case, to some inefficiency and confusion.

15 So I was wondering if Your Honor is open to relaxing  
16 that particular requirement in this case.

17 **MR. VAN NEST:** Your Honor, we don't think that would  
18 be appropriate. Both parties understood the rules and played  
19 by them. And they have the rebuttal case if they want it. And  
20 that's when the rebuttal ought to be.

21 **THE COURT:** Give me an example.

22 **MR. JACOBS:** So, for example, Your Honor, on the  
23 question of fair use, in our second expert report we addressed  
24 their opening on fair use in the nature of the copyrighted  
25 work. That's an example of something that would probably be

1 more efficient to put in on the opening report, even though --

2           **THE COURT:** No, I don't agree with that. I'm going  
3 to stick with my normal procedure.

4           Now, if you want to drop material from your opening  
5 report and skip over it, you don't have to put everything in  
6 that's in the opening report. But you do -- you should --  
7 we'll do it the way I like to lay it out. Then maybe fair use  
8 will drop out.

9           **MR. JACOBS:** Fair enough.

10          **THE COURT:** Or drop out in part.

11          **MR. JACOBS:** Understood.

12          **THE COURT:** All right.

13          **MR. JACOBS:** Second topic: Issues that have come up  
14 through direct testimony in the courtroom, our expert has been  
15 here for many of those days. He wasn't here today, but we can  
16 give him the transcript.

17           There are technical points that were not in their  
18 expert report so he couldn't put in his opposition even. But I  
19 think it would be helpful to the jury to hear our expert  
20 testify to things like the file cabinet or some of the specific  
21 issues that have come up as to what's in the Java language  
22 versus the API specification.

23           Of course, what's sauce for the goose would be sauce  
24 for the gander. We understand that. Dr. Astrachan would be  
25 similarly testifying based on what happens in the courtroom.

1                   **MR. VAN NEST:** Your Honor, we think that would be  
2 okay if Your Honor would allow it. Dr. Astrachan has been  
3 here, as well. And I would like to hear more about the file  
4 cabinet myself.

5                   (Laughter)

6                   **THE COURT:** Fine. If you both agree, that's good  
7 with me. Okay. Done.

8                   **MR. JACOBS:** Third is the question of word choice.  
9 Our expert is prepared to testify that it is his expert opinion  
10 that in the design of APIs, one engages in the creation of  
11 copyrightable expression using legal terminology.

12                  **THE COURT:** No, we don't do that.

13                  **MR. JACOBS:** I want to make sure we're on the same  
14 playing -- the same ground rules apply to both experts. He can  
15 testify to technical subjects. Frankly, I prefer he not  
16 testify to legal conclusions.

17                  **MR. VAN NEST:** I don't think you need my opinion on  
18 this one.

19                  **THE COURT:** Well, we don't have these kind of experts  
20 give legal opinions.

21                  **MR. VAN NEST:** That's right.

22                  **THE COURT:** All right. Sometimes it's a fuzzy line  
23 where you get to stop, but, no, the word "expression" that's  
24 too much. So we're not going to go there.

25                  **MR. JACOBS:** Thank you, Your Honor.

1                   **THE COURT:** I have some questions, but, Mr. Van Nest,  
2 let me give you your say.

3                   **MR. VAN NEST:** We don't have any issues, Your Honor.

4                   **THE COURT:** All right. Well, one of the issues I  
5 think I'm supposed to decide is the copyrightability of the 37  
6 APIs, among a few other things. So I would like to have some  
7 more briefing done on the following points.

8                   First of all, I want you to take a firm position.  
9 None of this business, the other side said. Your first  
10 reaction is always to say the other side says or the other side  
11 doesn't say. I want to know what you say, and I want a firm  
12 position.

13                  First is, could you get a patent on the structure,  
14 sequence, and organization? Assuming it meets novelty,  
15 assuming it's not anticipated, so forth, is that the proper  
16 subject matter of a patent?

17                  You should say "yes" or "no." Don't say, "We  
18 couldn't find any case law." Say "yes" or "no," and "here's my  
19 reason why."

20                  All right. Second, does the Copyright Office  
21 investigate the SSO? I don't think so. I think they would be  
22 amazed to find out that you were contending that they had  
23 blessed the structure, sequence, and organization, because I  
24 bet you they don't even know what it is, it's so hard to figure  
25 out. But maybe I'm wrong.

1           I would like to know, when somebody registers  
2 something with the Copyright Office is that -- is the  
3 structure, sequence, and organization of the source code  
4 investigated by the Copyright Office?

5           I bet you can answer that right now. What's the  
6 answer?

7           **MR. BABER:** Your Honor, the answer is, no, there is  
8 no investigation. In fact, in support of the registration they  
9 don't even have to send the Copyright Office a complete copy of  
10 the whole work. With software all they have to send in is the  
11 first 50 and last 50 pages of code. So the Copyright Office  
12 doesn't even have the work, let alone be in a position to make  
13 any kind of examination or assessment.

14           **THE COURT:** Isn't that right?

15           **MR. JACOBS:** I think that's probably mostly right.

16           **THE COURT:** What part is not right?

17           **MR. JACOBS:** Well, what the Copyright Office does is  
18 examine the materials that are submitted for registration, for  
19 registerability. If the Copyright Office saw on its face that  
20 you were seeking to register something that was not eligible  
21 for copyright protection, your registration certificate would  
22 be rejected.

23           In this particular case, the submission to the  
24 Copyright Office would include not only whatever source code  
25 was submitted -- and that may depend on the particular

1 registration in question -- but the documentation, as well.

2 So --

3           **THE COURT:** Is there something there that says, Heads  
4 up, Copyright Office, we're also seeking copyright protection  
5 on the structure, sequence, and organization?

6           **MR. JACOBS:** No, there is no such heads up, Your  
7 Honor.

8           **THE COURT:** How do they know they are blessing that  
9 theory?

10          **MR. JACOBS:** I don't think there's any -- I don't  
11 think there's any imprimatur on the theory of copyrightability.  
12 I think there is an imprimatur on the registerability of the  
13 underlying material.

14          **THE COURT:** I -- okay. This piece just goes to how  
15 much weight do we give to the fact that it was registered  
16 when -- you know, the words, okay. Like if you submit a poem  
17 and they give you a copyright on a poem. Everybody knows it's  
18 those words in that order. And somebody else could come along  
19 with a different poem on the same subject but it might have  
20 some of the same words. But if it was, you know, reorganized  
21 it would not be plagiarism.

22          Anyhow, all right. So then the other question is  
23 this: I have one last question. Derivative work. In order to  
24 prove something is a derivative work, does the plaintiff have  
25 to prove that the defendant actually used the copyrighted work

1 in order to produce the derivative work?

2                   Or what if they just come up with their own -- their  
3 own words, but they happen to be somewhat similar but not  
4 identical?

5                   In other words, for a derivative work, not for the  
6 copying, direct copying, but for derivative works, does the  
7 plaintiff have to prove that the defendant actually had in  
8 their possession and worked off of and derived their own work  
9 from the copyrighted work?

10                  What's the answer to that?

11                  **MR. JACOBS:** If I can step back for a second, I think  
12 the point about derivative work -- well, let me answer your  
13 question directly --

14                  **THE COURT:** Sure.

15                  **MR. JACOBS:** -- and give you the explanation I was  
16 about to give you.

17                  The short answer is that, as with an infringing work  
18 which is of the same form as the copyrighted work, the same is  
19 true for a derivative work. The plaintiff shows access and  
20 substantial similarity, and that allows an inference of  
21 copying. And then it's up to the defendant to rebut that based  
22 on some theory of independent development. So access and  
23 substantial similarity is the method of proof, regardless of  
24 the form of the work.

25                  I think the only reason we're into derivative work

1 here is that we're talking about, arguably, a recasting of the  
2 documentation in a different form, i.e. computer program code.

3 And so if the -- if what Google is claiming it did is  
4 take the documentation and transform it or recast it into a  
5 computer program, then it's not documentation to document  
6 copying. It's documentation into computer program copying.  
7 And that fits the derivative work schema.

8 But, actually, the basic proof remains substantial  
9 similarity of copyrightable material.

10 **THE COURT:** What does Google say on that point?

11 **MR. BABER:** Your Honor, no surprise, we disagree. We  
12 think that when you're talking about a derivative work -- I  
13 believe the Ninth Circuit requires that a derivative work, to  
14 be an infringement, has to include protectable elements,  
15 copyrightable elements of the first work.

16 Okay. So in that respect, yes, Google would have had  
17 to have used something from the Java materials in the APIs that  
18 was protectable by copyright, okay, for it to be an  
19 unauthorized derivative work that infringes.

20 **MR. JACOBS:** I don't think we disagree with that,  
21 Your Honor. That's essentially what I said.

22 **THE COURT:** All right. Well, let me ask a fact  
23 question to Google. What is the evidence going to show on  
24 this? Will the evidence show that in a clean room the Google  
25 engineers who were purporting to come up with the Android

1 source code had before them the English language comments in  
2 the Oracle documentation?

3           **MR. BABER:** There was (unintelligible) in discovery,  
4 Your Honor, that at least some of the engineers who worked on  
5 the libraries did have access to, did look at, some of the  
6 English language prose descriptions of the APIs.

7           **THE COURT:** And was that copyrighted, the English  
8 language prose?

9           **MR. BABER:** Well, you honestly can't tell what's  
10 copyrighted, Your Honor. They were on a website where there's  
11 a copyright notice on every page.

12           It's similar to the copyright registration. A big  
13 platform, you never have to say what's copyrighted and what's  
14 not.

15           **THE COURT:** Let's assume it is copyrighted. Would  
16 that then be a derivative work, the source code that was  
17 created based upon the plain English thing? Assuming it was  
18 copyrighted, would that then be a derivative work?

19           **MR. BABER:** No, Your Honor, I don't believe so.  
20 Because if you're talking about just the English prose  
21 descriptions -- okay, we're not talking about the method  
22 signatures, the package names, any of that -- if you're talking  
23 about just the English prose descriptions, which are like the  
24 definitions in a dictionary, then you would have to look at  
25 what are the corresponding English prose descriptions in the

1 Google libraries. And you would have to compare all the  
2 dictionary entries in one to all the dictionary entries in the  
3 other to compare the two works as a whole.

4           **THE COURT:** Well, all right. I see your point. Even  
5 there, though, even English commentary to English commentary,  
6 there are -- it's not identical, but there are a number of  
7 places where it's pretty close. What's your answer going to be  
8 to that?

9           **MR. BABER:** The answer to those, Your Honor, is that,  
10 first of all, this is very technical language.

11           And if you look at them, they are all written in  
12 pretty much the same style. The Java language dictates some of  
13 that: This API returns X. Well, that's what you would expect  
14 the description of the API to be. It returns X.

15           So on those that's, Your Honor, when we briefed  
16 previously some of the other copyright doctrines that are out  
17 there. The merger doctrine, when you have a very short  
18 sentence that is very simple. There have been copyright cases  
19 "apply hook to wall" --

20           **THE COURT:** I understand. But we've got hundreds of  
21 these, and some of them are longer than four words. There are  
22 some of them that are 12 or 15 words long, and they kind of  
23 read the same. Don't you think?

24           **MR. BABER:** No, Your Honor, we don't. And that's not  
25 our expert's view either.

1           If you look at them and look at lots of pairs of  
2 them, do they use the same key words? Absolutely. Because  
3 they are describing the same API that takes the same arguments  
4 as input and takes the same return as an output. And you have  
5 to use those words. You have to put in X to get out Y. And  
6 this is what the API does for you.

7           But if you -- there's lots -- we have several  
8 exhibits that were used in connection with the summary judgment  
9 motion, as I recall, that shows the descriptions are, in many  
10 cases, very, very, very different, even though they are  
11 describing the same thing.

12           And the similarities, generally, are a result of the  
13 fact that they are describing the same thing.

14           **THE COURT:** Go back to your earlier point about  
15 English language commentary and whether or not the source code  
16 by Google is a derivative work.

17           The analogy is, I think -- I'm trying to make sure I  
18 understand it. The analogy would be to those cases that say  
19 you describe how the game is played, but that doesn't -- gives  
20 you a copyright on that description, but it doesn't give you a  
21 copyright on playing the game.

22           **MR. BABER:** Even worse than that, Your Honor. Yes,  
23 correct. That and the old *Baker vs. Selden* case, about the  
24 accounting, the double-entry bookkeeping method case, which  
25 says, if you teach someone how to do something, that doesn't

1 give you a monopoly on them doing it.

2           But if you think about the argument they're making  
3 right now, imagine a small API, an API that the English prose  
4 description says "returns the square root of a number." Okay.  
5 We're now in a world where the method name is not protectable.  
6 The package name is not protectable. The class name is not  
7 protectable. What they're saying is, look, you created source  
8 code that returns the square root of a number; therefore, you  
9 infringe.

10           Well, Your Honor, it would be copyright law to the  
11 extreme to say that a program, any program that calculates a  
12 square root infringes someone's rights in a sentence that says  
13 "returns the square root of a number" as one of thousands and  
14 thousands of comparable statements.

15           **THE COURT:** Well, it's more to it than that.

16           They say that you have organized your methods and  
17 fields into the same set of classes, into the same set of  
18 packages that -- for these 37 that the -- that they organized  
19 their's. So it's more than just one square root thing.  
20 It's -- I've forgotten the number -- 7,000 methods that are all  
21 organized exactly the same way. I think that's -- that's their  
22 structure, sequence, and organization argument.

23           You don't have to answer that. I think that's  
24 their -- that's their point. And that's one of the issues that  
25 have to be decided, I think, by the judge here. Right?

1                   **MR. BABER:** Yes, I believe -- well, as to whether or  
2 not those things are copyrightable, absolutely, we think that's  
3 your job, Your Honor.

4                   If I can just make two comments. I think you heard  
5 testimony today from Dr. Bloch, about APIs, that once you give  
6 an API a name, that's your structure. That's your organization  
7 under the language specification. You've told it what package  
8 it's in, what class it's in. You've given it a name as a  
9 method or something else that's in the package. And that's the  
10 organization. The names are the organization under the *Java*  
11 *Language Specification* for these so-called fully-qualified  
12 names.

13                  There isn't a separate structure, sequence, and  
14 organization that is somewhere in the world apart from the  
15 names. It is the names that put them in those packages and in  
16 those classes.

17                  So -- and I want to make sure we're on the same page.

18                  **THE COURT:** Couldn't you have -- under your  
19 organization, you could have put the square root thing, instead  
20 of putting it under math, you might have put it under science.  
21 And you didn't have to use the exact same setup and  
22 organization that Java did.

23                  **MR. BABER:** Your Honor, but if you did, if you put it  
24 somewhere else, two things. First of all, any existing code  
25 that used that API wouldn't work because it would go to find

1 that code to perform the function, and it wouldn't be where  
2 it's called, number one.

3                 And, number two, all those programmers that you heard  
4 about, who've not only learned these APIs but memorized them so  
5 that they can write their programs using this shorthand, using  
6 this special vocabulary, those people would have to learn a new  
7 vocabulary.

8                 So that wouldn't work if you gave it a different  
9 name, and it was in a different package or a different class.

10               **THE COURT:** Okay. One last question. And that is,  
11 are there subsidiary questions of fact that the jury should  
12 decide, that would tie into the judge's determination on  
13 copyrightability? Or is it all just up to the judge to make  
14 his or her own decision on the facts?

15               **MR. BABER:** I believe, Your Honor, it's up to you.  
16 And, memory serves, in our briefs we have cited cases where  
17 other District judges have said the jury has no role to play in  
18 deciding issues of copyrightability.

19               **THE COURT:** Let's hear from Mr. Jacobs. What do you  
20 say on that issue?

21               **MR. JACOBS:** I think I still am where I was on this,  
22 which is, depending on where you draw that line, you could  
23 create subsidiary issues of fact.

24               **THE COURT:** Give me an example.

25               **MR. JACOBS:** Well, suppose the Court were to say that

1 it -- it would be arbitrary. I think both sides are saying to  
2 you that the structure, sequence, and organization is the  
3 design of the APIs and their organization. And Google is  
4 saying to you that's uncopyrightable, and we're saying to you  
5 that's copyrightable.

6 And if you instruct the jury that it is -- that the  
7 structure, sequence, and organization, as reflected in the  
8 design of the 37 packages that are at issue in this lawsuit, is  
9 protectable subject matter under the copyright laws, it is  
10 copyrightable, then I think we have not only -- then I think we  
11 have infringement because they've essentially conceded -- they  
12 have conceded substantial similarity.

13 And then the only question is, do they have an excuse  
14 for that infringement based on this desire for compatibility  
15 that Mr. Baber just referred to?

16 But an arbitrary line, for example, that said, well,  
17 the names on the left side, they're not protectable, so you've  
18 got to compare the names in the middle and the names on the  
19 right, that's some kind of -- some kind of odd instruction that  
20 I don't think either of us are going to ask you for, could  
21 create fact issues.

22 **THE COURT:** Well, on the fair use defense, let's  
23 assume that it is copyrightable for a moment. Is fair use for  
24 the judge or for the jury?

25 **MR. JACOBS:** I think here it will ultimately be for

1 the judge. But it is -- but fair use, at least, unlike  
2 copyrightability, is an issue with either a factual issue or an  
3 issue with facts underlying it.

4           **THE COURT:** Why is it for the judge?

5           **MR. JACOBS:** Because I think, as a matter of law,  
6 given the commercial use that Google is making of these -- of  
7 these API designs, there can be no question of fair use.

8           They'll disagree with that, but that is an issue we  
9 would brief to you as a JMOL motion pre-verdict, if you were to  
10 instruct that the design of the API as reflected in the  
11 structure, sequence, and organization is subject copyrightable  
12 subject matter.

13           **THE COURT:** Let's assume that that's what the court  
14 decides, that you're right that it's copyrightable. And I'm  
15 not -- I'm just saying arguendo, assuming arguendo that.

16           You're saying that the jury should not be allowed to  
17 even consider fair use because the asserted reasons for fair  
18 use are invalid as a matter of law?

19           **MR. JACOBS:** Yes. And we would so move.

20           **THE COURT:** Have you briefed that yet?

21           **MR. JACOBS:** No. That has not been -- well,  
22 actually, in our summary judgment motion we took this on, I  
23 believe.

24           **THE COURT:** Well -- all right.

25           **MR. JACOBS:** But not in the way -- not in the level

1 that you've had us briefing in the last few weeks, Your Honor.

2           **THE COURT:** Sorry you have to work so hard.

3           **MR. JACOBS:** No, it's good.

4           **THE COURT:** What do you say on the Google side about  
5 that, those issues?

6           **MR. BABER:** Well, no surprise, I couldn't disagree  
7 more with Mr. Jacobs about one thing, which is his last point.

8           The Supreme Court has made clear more than once that  
9 no one of the four statutory fair use factors is dispositive.  
10 Period.

11           **THE COURT:** What are they?

12           **MR. BABER:** The nature of the copyrighted work, the  
13 amount taken in reference to the work as a whole, and the  
14 purpose of the work, and the affect on the market, or potential  
15 market for the copyrighted work, and any other factors that are  
16 deemed relevant. So in this case --

17           **THE COURT:** If it goes to the jury, is that what the  
18 Supreme Court says?

19           **MR. BABER:** I'm not sure the Supreme Court has said  
20 it goes to the jury. I think it does go to the jury. I'm not  
21 sure it's a Supreme Court case that says it goes to the jury,  
22 but it's a statutory defense under the copyright statute. So  
23 just like infringement, I think fair use goes to the jury.

24           **THE COURT:** So your argument would be -- let me see  
25 if I can say your argument, so I can focus for a moment.

1           Your argument would be, all right, assume that  
2 structure, sequence, and organization is copyrightable.  
3 Nonetheless, Google spent much time and effort to write up its  
4 own source code, and while it does use the same structure,  
5 sequence, and organization on the 37 APIs, out of -- I'm making  
6 this part up -- out of 17,000 lines of code only 1,000 are the  
7 same.

8           **MR. BABER:** 15 million, Your Honor.

9           **THE COURT:** 15 million, I mean.

10           (Laughter)

11           **THE COURT:** Only a few hundred are the same. And  
12 those are the declarations.

13           And so your argument would be that that small amount  
14 of copying of the outline is fair use in order to achieve the  
15 public interest of compatibility.

16           **MR. BABER:** That's my second argument, Your Honor.

17           **THE COURT:** What's your first argument?

18           **MR. BABER:** My first argument is, even if it's  
19 copyrightable, there is still no infringement because the  
20 elements that were taken are a relatively small part of the  
21 work that they registered, which includes not just these 37  
22 APIs and the other 14 APIs we have, that they've dropped during  
23 the course of the lawsuit, but it also includes 110 other APIs,  
24 and it includes a virtual machine and other things.

25           So whenever you're talking about infringement, the

1 Ninth Circuit has said over and over again you have to compare  
2 the two works as a whole. You don't take one file to one file.  
3 You don't take one name to one name. You take, what did the  
4 plaintiff register? What is its work?

5 Their work is a very big work. They say Android  
6 infringes. Android is a very big work. So when you compare  
7 the whole Java platform to Android, we believe the jury could  
8 easily make a factual finding, okay, there are some parts that  
9 are similar or the same, but no one would ever say that  
10 Android, as a work, was a copy of the Java platform. And  
11 that's the appropriate test for infringement.

12 So that's my first argument.

13 **THE COURT:** All right.

14 **MR. BABER:** Then my second argument is, even if it  
15 wasn't infringement, then, yes, this was clearly a  
16 transformative use of those limited elements of the APIs. The  
17 names, the method declarations, those limited elements out of  
18 the large Java work have been used in a transformative way to  
19 come up with a much larger, much bigger, much more complex  
20 work.

21 And transformativeness, the Supreme Court has said  
22 recently, is a critical element in terms of whether or not a  
23 use is a fair use.

24 **THE COURT:** What is your witness or witnesses who are  
25 going to address fair use?

1                   **MR. BABER:** Your Honor, fair use will be addressed  
2 through a number of witnesses.

3                   Our technical people will talk about just like --  
4 well, Dr. Bloch, didn't get to it today, but they'll talk about  
5 what's actually in Android. In other words, what parts of the  
6 Java APIs are actually in Android, the method signatures. Our  
7 expert will certainly talk about the same thing, Dr. Astrachan.  
8 Dr. Astrachan can certainly talk about the nature of the work,  
9 the Java platform. It is a software work. It's a functional  
10 work. It's utilitarian. Et cetera. In terms of trying to  
11 stick to the factors in my head.

12                  Transformativeness, certainly Mr. Rubin and  
13 Mr. Bornstein and other people who have worked on Android can  
14 talk about this is what Android is, the whole thing, what the  
15 platform is, what it does. Their testimony will all go to the  
16 issue of transformativeness.

17                  **THE COURT:** Is that a Supreme Court word?

18                  **MR. BABER:** Transformative?

19                  **THE COURT:** Yeah.

20                  **MR. BABER:** I believe it is, Your Honor. But,  
21 certainly a Ninth Circuit word. That's for sure.

22                  **THE COURT:** All right. Do you have any internal  
23 e-mails that say this is fair use? Or is that something that  
24 is after the fact?

25                  **MR. BABER:** Your Honor, I don't think we have any

1 internal e-mails that say it's fair use or talk about  
2 structure, sequence or organization, or talk about any kind of  
3 copyright concepts.

4 Your Honor, if I may, I want to exclude from my last  
5 answer: Obviously, I have not reviewed anything that's  
6 privileged, that would come from the Google legal department.

7 **THE COURT:** I understand that. Possibly it's  
8 privileged, but none of us will see that.

9 Well, the five questions that I gave you today do not  
10 get to fair use, and so forth. But I would be interested in  
11 getting tomorrow your first cut at how we can pose the  
12 questions to the jury, including, you know, you can write up  
13 a -- red line all you want. But I would like you to do it in  
14 handwriting so it will be clear to me where you want to make  
15 the changes. But then you can add the additional questions  
16 about fair use so I can see how that will fit in.

17 What I'm trying to do before I draft the jury  
18 instructions, it's always best to draft the verdict form. Or  
19 get a good draft of it. Because that, in turn, helps me  
20 understand what I need to say to the jury.

21 You can help me greatly if any of you would  
22 streamline things by dropping issues. That way, I don't have  
23 to instruct on things that may wind up being a point of  
24 confusion.

25 But, you know, I can't require you to drop anything.

1 I'm just saying, sometimes good lawyers like you will realize  
2 that dropping side shows may help you in the long-run.

3 All right. I've run out of things. This has been a  
4 useful conversation for me.

5 **MR. VAN NEST:** Your Honor.

6 **THE COURT:** Yes.

7 **MR. VAN NEST:** Just to clarify what you want, I  
8 understand you want the jury questions you gave us this morning  
9 red lined. You want them in the morning.

10 **THE COURT:** Yes.

11 **MR. VAN NEST:** And then you mentioned you had some  
12 questions you wanted us to brief, that we've been talking  
13 about.

14 **THE COURT:** You can have a little longer on that.  
15 When do you think you can get me the briefs? Maybe by Sunday  
16 afternoon?

17 **MR. VAN NEST:** I would think so. Sunday afternoon.

18 **THE COURT:** Why are you looking around?

19 (Laughter)

20 **MR. VAN NEST:** Sunday afternoon. I don't know why  
21 I'm always the one, Your Honor. But Sunday afternoon what  
22 time?

23 **THE COURT:** 3:00 p.m.

24 **MR. VAN NEST:** Fair enough. We'll do it. Page  
25 limit? The sky's the limit?

1 (Laughter)

2 **THE COURT:** How about ten pages. Will that work?

3 **MR. JACOBS:** That's fine, Your Honor.

4 **MR. VAN NEST:** We'll do that.

5 **THE COURT:** Fine. See you tomorrow.

6 **MR. VAN NEST:** Thank you, Your Honor.

7 **MR. JACOBS:** Thank you, Your Honor.

8 (At 1:35 p.m. the proceedings were adjourned until  
9 Friday, April 20, 2012, at 7:30 a.m.)

10 - - - - -

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

I N D E X			
	<u>PLAINTIFF'S WITNESSES</u>	<u>PAGE</u>	<u>VOL.</u>
1	<b>REINHOLD, MARK B.</b>		
2	(PREVIOUSLY SWORN)	670	4
3	Direct Examination Resumed by Mr. Jacobs	670	4
4	Cross Examination by Mr. Purcell	698	4
5	Redirect Examination by Mr. Jacobs	727	4
6			
7	<b>BLOCH, JOSHUA</b>		
8	(SWORN)	731	4
9	Direct Examination by Mr. Jacobs	732	4
10	Cross Examination by Mr. Baber	757	4
11	Redirect Examination by Mr. Jacobs	827	4
12	Recross Examination Resumed by Mr. Baber	832	4
13	Further Redirect examination by Mr. Jacobs	836	4
14			
15			
16	- - -		
17			
18			
19			
20			
21			
22			
23			
24			
25			

1  
**E X H I B I T S**  
2

	<u>TRIAL EXHIBITS</u>	<u>IDEN</u>	<u>VOL.</u>	<u>EVID</u>	<u>VOL.</u>
3	3, 9, 10, 11, 12			838	4
4	17, 125, 212, 213, 222			838	4
5	25, 987			841	4
6	30			656	4
7	36, 37, 38, 39, 40			655	4
8	41, 42, 43, 44, 45, 46			655	4
9	47, 51, 53, 54, 65, 70, 74			655	4
10	317, 318, 321, 325, 330			838	4
11	326			852	4
12	610.2			672	4
13	622			692	4
14	623			691	4
15	623.1 - 623.10			695	4
16	624			738	4
17	794			755	4
18	965			692	4
19	980, 981			690	4
20	1062, 1063			681	4
21	3452	834	4		
22	4027			835	4
23		— — —			
24					
25					

1  
2  
3 **CERTIFICATE OF REPORTERS**

4 We, KATHERINE POWELL SULLIVAN and DEBRA L. PAS,  
5 Official Reporters for the United States Court, Northern  
6 District of California, hereby certify that the foregoing  
7 proceedings in C 10-3561 WHA, **Oracle America, Inc., vs. Google,**  
8 **Inc.**, were reported by us, certified shorthand reporters, and  
9 were thereafter transcribed under our direction into  
10 typewriting; that the foregoing is a full, complete and true  
11 record of said proceedings at the time of filing.

12

13 /s/ Katherine Powell Sullivan

14

15 Katherine Powell Sullivan, CSR #5812, RPR, CRR  
U.S. Court Reporter

16

17

18 /s/ Debra L. Pas

19

Debra L. Pas, CSR #11916, RMR CRR

20

21

Thursday, April 19, 2012

22

23

24

25

*Katherine Powell Sullivan, CSR, CRR, RPR  
Debra L. Pas, CSR, CRR, RMR  
Official Reporters - US District Court - 415-794-6659*